UNIVERSITY OF OSTRAVA
FACULTY OF SCIENCE
DEPARTMENT OF MATHEMATICS

FUZZY TRANSFORM AND ITS APPLICATIONS TO PROBLEMS IN ENGINEERING PRACTICE

Ph.D. THESIS

AUTHOR: Martin Štěpnička
SUPERVISOR: Irina Perfilieva

2007

"If a man will begin with certainties,
he shall end in doubts;
but if he will be content to begin with doubts
he shall end in certainties."

Francis Bacon

Summary

The fuzzy set theory initiated by L.A. Zadeh provided mathematicians with an appropriate tool for modelling the vagueness phenomenon and shed new light into the control theory for engineers. Later, in 1985, T. Takagi and M. Sugeno invented a particular fuzzy model which became very popular due to its approximation ability. Finally, in the 1990's, several studies aimed at approximation properties of the other widely used fuzzy models.

Based on the historical development briefly recapitulated above, a new field called *fuzzy approximation* focusing on approximation properties of fuzzy models and development of new methods using fragments of the fuzzy set theory has been established. Fuzzy transform (F-transform), a particular method belonging to this field, is the main object of the study in this thesis.

Fuzzy transform has been proposed as a pilot fuzzy approximation technique with the aim of being applied in up to now unusual application fields such as numerical solution of differential equations, for example. On the other hand, such techniques, including the fuzzy transform, are not excluded from the other techniques involved in fuzzy systems. Vice-versa, if they are correctly built (i.e. if they respect rules of law of fuzzy logics and the state of arts in fuzzy system), they provide us with powerful tools for dealing with typical problems for implementations of fuzzy system.

The goal of this thesis is to investigate the fuzzy transform from the approximation point of view and to incorporate it in further numerical methods. In general, we talk about numerical methods on the basis of fuzzy approximation models. Moreover, the technique is studied in the context of other fuzzy models and finally implemented in automatic control, a typical field for fuzzy approaches.

The structure of the work is as follows. Chapter 1 provides an introduction to the study and a brief state of the art of the studied fields. Chapter 2 recalls basic definitions and facts about the F-transform method and introduces new results useful for further chapters. Chapter 3 focuses on an extension of the F-transform for function with two or more variables and presents analogous results to those from the one dimension. In Chapter 4, a possible application to numerical computation

of partial differential equations is studied.

Chapter 5 is devoted to the so-called additive normal forms. It introduces a class of additive normal forms stemming from the one defined in [59]. A representation of the F-transform as a special case of the additive normal form, its extension to a normal form with other possible operations and answering natural questions about the relations between F-transform and other fuzzy approximation techniques is the main aim of the chapter.

Chapter 6 introduces additive interpretations of fuzzy rule bases and aims at their fuzzy interpolation properties. It presents a possible way how to identify a rule base with the additive interpretation using the F-transform. Chapter 7 is the application part of the Thesis which deals with a heuristic proposal for fuzzy control stemming from additive normal forms and the F-transforms. This chapter explicitly uses results from the previous chapters to demonstrate properties of the proposed method. Good behaviour of a system controlled by the proposed method is demonstrated on a real application: control of an autonomous robot.

Finally, Chapter 8 provides a neural network point of view to the F-transform and presents experiments justifying this approach. The last chapter summarizes the results from the thesis and briefly discusses them.

Keywords: Fuzzy transform, Approximation, Numerical methods, Partial differential equations, Normal forms, Fuzzy rule based systems.

Anotace

Teorie fuzzy množin iniciována L.A. Zadehem poskytla matematikům vhodný nástroj k modelování fenoménu vágnosti a inžnýrům vnesla nové světlo do teorie řízení (regulace). Později, v roce 1985, T. Takagi a M. Sugeno navrhli fuzzy model, který se stal oblíbený díky svým aproximačním možnostem. Konečně v 90-tých letech se několik prací zabývalo aproximačními možnostmi dalších obecně používaných fuzzy modelů.

Na základě tohoto historického vývoje byly položeny základy nového oboru zvaného fuzzy aproximace, který se zabývá aproximačními vlastnostmi fuzzy modelů a jejich rozvojem. Fuzzy transformace (F-transformace) jakožto konkrétní zástupce metod tohoto oboru je hlavním objektem studia této práce.

Fuzzy transformace byla navržena jako pilotní fuzzy aproximační technika s cílem být aplikována v dosud netradičních oblastech jako nařiklad numercické řešení diferenciálních rovnic. Na druhou stranu, takové techniky nestojí mimo ostatní techniky implementované ve fuzzy systémech. Naopak, pokud jsou korektně postaveny tj. respektují pravidla a zákonitosti fuzzy logik a fuzzy systémů, poskytují nám mocné nástroje pro práci s problémy typickými pro implementaci fuzzy systémů.

Cílem této práce je výzkum fuzzy transformace z aproximačního hlediska a její použití v numerických metodách. Obecně pak mluvíme o numerických metodách na základě fuzzy aproximačních modelů. Dále je technika studována v kontextu ostatních fuzzy modelů a nakonec implementována v automatickém řízení tj. oblasti velmi typické pro fuzzy přístupy.

Struktura práce je následující. Kapitola 1 je úvodem do studia a úvodem do oblastí uvažovaných v práci. Kapitola 2 připomíná základní definice a fakta o metodě F-transformace a uvádí nové výsledky použité v dalších kapitolách. Kapitola 3 se zabývá rozšířením F-transformace na funkce dvou a více proměnných a uvádí výsledky analogické jednodimenzionálnímu případu. V kapitole 4 a je studována možnost aplikace metody na numerický výpočet parciálních diferenciálních.

Kapitola 5 je věnována tzv. aditivním normálním formám. Představuje třídu aditivních normálních forem motivovanou konkrétním případem publikovaným v

[59]. Reprezentace F-transformace jako specielního případu aditivní normální formy, její rozšíření pro další operace a studium přirozených otázek o vztahu F-transformace a dalších fuzzy aproximačních technik je hlavním přínosem kapitoly.

Kapitola 6 představuje aditivní interpretace bází fuzzy pravidel a zaměřuje se na jejich fuzzy interpolační vlastnosti. Možný způsob identifikace báze s takovou interpretací využívající F-transformaci je zde uveden také. Kapitola 7 je aplikační částí práce, která pracuje s heuristickým návrhem pro fuzzy regulaci vycházejícím z aditivních normálních forem a F-transformace. Tato kapitola přímo používá výsledky předchozích kapitol a demonstruje vlastnosti navrhované metody. Dobré chování procesu řízeného navrhovanou metodou jsou demonstrovány na reálné aplikaci - na řízení autonomního robota.

Konečně, kapitola 8 uvádí přístup k F-transformaci z pohledu neuronových sítí. Poslednǐ kapitola shrnuje výsledky práce.

Klíčová slova: Fuzzy transformace, Aproximace, Numerické metody, Parcilální diferenciální rovnice, Normální formy, Systémy s bází fuzzy pravidel.

Preface

The fuzzy set theory provides us with tools for modelling meanings of linguistic expressions and modelling imprecise rough and vague evaluations and deal with them. The fuzzy logic is a special many-valued logic serving for the vagueness phenomenon involvement. Sometimes, the notion *fuzzy logic* is understood much wider and we then talk about so-called *fuzzy logic in broader sense (FLb)*, while the genuine many-valued logic is called *fuzzy logic in narrow sense (FLn)* [52].

Let us stress that imprecision is essential in real life and precise measurements are only illusions caused by rounded values. Even in the case of the latest measuring apparatuses we cannot achieve precision without being on the level of the micro-world of molecules and atoms. We should also stress that besides the fact that inaccuracies and imprecisions are unavoidable, human language and human way of understanding always deal with this feature. Everyone who was trained to drive a car was taught by sentences containing vague expressions such as: "turn to the left *a little*" or "slow down *gently*". One could hardly expect that teaching by instructions such as: "turn the wheel to the right up to $24°16'42''$ and slow down by pushing the break pedal with strength of 12.681679234 N" would lead to a fast and successful result.

The principle described on the driving example above is nicely formulated in the so-called **Principle of Incompatibility** [79]:

> *As the complexity of a system increases, human ability to make precise and relevant (meaningful) statements about its behaviour diminishes until a threshold is reached beyond which the precision and the relevance become mutually exclusive characteristics.*

Moreover, we can find ancient roots of this idea already in the **Socrates Paradox**:

The less we know, the more certain and precise we are in our explanations; the more we know, the more we realize the limitations of being certain and precise.

Therefore, the study of fuzziness of human knowledge - its sources, nature and dynamics - is not motivated by an effort to reduce or eliminate it but to understand its limitations and to learn how to deal with it.

This thesis is devoted to fuzzy approximation methods and their applications. Particularly, it focuses on the fuzzy transform (F-transform) [58] technique and investigates it from the point of view of numerical mathematics of other fuzzy approximation methods and neural networks. Furthermore, it aims at possible applications of the technique.

I want to express my gratitude to my supervisor Prof. Irina Perfilieva for her support, valuable comments and permanent encouragement which made it possible to finish this thesis. Moreover, I would like to thank my boss Prof. Vilém Novák for working conditions he created in the Institute for Research and Applications of Fuzzy Modeling. Warm thanks goes to all my present and former colleagues form the institute for making it a friendly and creative atmosphere. Especially, I would highlight Martina Daňková for her co-operation and for the careful reading of this thesis, Radek Valášek for a fruitful and long co-operation and finally Ondřej Polakovič and Viktor Pavliska for their experimental and software support.

Ostrava, March 2007 Martin Šěpnička

Table of Contents

List of Figures

List of Symbols

Chapter 1

Preliminaries

1.1 Introduction to Fuzzy Set Theory

Let us briefly recall the basic elements of the fuzzy set theory which development has been initiated by L.A. Zadeh [80].

1.1.1 Fuzzy Sets

A classical (crisp) set, say A, is such a collection of objects from some domain (universe), say $\mathcal{D}$, that we are able to distinguish whether an arbitrary considered object from $\mathcal{D}$ belongs to A or does not belong to A. Objects x belonging to A are called elements of A, which is denoted by $x \in A$. Set A is a subset of the domain which is denoted by $A \subseteq \mathcal{D}$. Any classical set can be characterized by its characteristic function $\chi_A : \mathcal{D} \to \{0,1\}$

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

Fuzzy sets allow the whole interval $[0,1]$ to be the range of their characteristic functions which we call *membership functions*. We unify a fuzzy sets $\mathbf{A}$ with its membership function $\chi_{\mathbf{A}}$ and consider $\mathbf{A} : \mathcal{D} \to [0,1]$ denoted by $\mathbf{A} \underset{\sim}{\subseteq} \mathcal{D}$. The value $\mathbf{A}(x)$ for $x \in \mathcal{D}$ is called the *membership degree* of x to $\mathbf{A}$. Immediately, each classical set is a special case of a fuzzy set and therefore the fuzzy set do not contradict classical sets but generalize them.

The definition of a fuzzy set allows us to model vague human language notions. Especially, so called evaluating linguistic expressions (*small, very big, more or less* *medium, about five, etc.*) [48, 50] can be successfully modelled by fuzzy sets, see Fig. 1.2.

Definition 1 Let $\mathbf{A} \underset{\sim}{\subseteq} \mathcal{D}$. Then the *support* of $\mathbf{A}$ is the following set

$$\mathrm{Supp}(\mathbf{A}) = \{x \mid x \in \mathcal{D},\ \mathbf{A}(x) > 0\}. \tag{1.1.1}$$

Definition 2 Let $\mathbf{A} \underset{\sim}{\subseteq} \mathcal{D}$. Then the *kernel* of $\mathbf{A}$ is the following set

$$\mathrm{Ker}(\mathbf{A}) = \{x \mid x \in \mathcal{D},\ \mathbf{A}(x) = 1\}. \tag{1.1.2}$$

The support is a set of elements from a domain with a non-zero membership degree to a given fuzzy set or a set of those elements od $\mathcal{D}$ which at least partially belong to the fuzzy set. The kernel is a set of elements of $\mathcal{D}$ which fully belong to the fuzzy set. A kernel can be an empty set while a support can be empty only in the case of so called *empty fuzzy set* - a fuzzy set which contains all elements of $\mathcal{D}$ with a zero membership degree.

Definition 3 Let $\mathbf{A} \underset{\sim}{\subseteq} \mathcal{D}$ and $\alpha \in [0,1]$. Then the set

$$\mathbf{A}_\alpha = \{x \mid x \in \mathcal{D},\ \mathbf{A}(x) \geq \alpha\} \tag{1.1.3}$$

is called the *α-level set* (sometimes α-cut).

Immediately, we get that the kernel can be defined as $\mathbf{A}_\alpha$ for $\alpha = 1$.

Remark 1 *Some authors also define α-level sets as follows*

$$\mathbf{A}_\alpha = \{x \mid x \in \mathcal{D},\ \mathbf{A}(x) > \alpha\}.$$

In this case, the support can be defined as $\mathbf{A}_0$. Let us stress, that we will follow the terminology from Definition 3 in the latter.

For α-level sets the following condition holds:

$$\text{If} \quad \alpha \leq \beta \quad \text{then} \quad A_\beta \subseteq A_\alpha. \tag{1.1.4}$$

The following representation theorem claims that a fuzzy set can be represented by its α-level sets.

Theorem 1 *Let $\mathbf{A} \underset{\sim}{\subseteq} \mathcal{D}$. For any element $x \in \mathcal{D}$ its membership degree to $\mathbf{A}$ can be expressed as follows*

$$\mathbf{A}(x) = \bigvee_{x \in \mathbf{A}_\alpha} \alpha. \tag{1.1.5}$$

Definition 4 $\mathbf{A} \underset{\sim}{\subseteq} \mathcal{D}$ is called *convex* if for any $x, y \in \mathcal{D}$ and for any $\lambda \in [0, 1]$ the following formula holds

$$\mathbf{A}(\lambda x + (1 - \lambda) y) \geq \mathbf{A}(x) \wedge \mathbf{A}(y). \tag{1.1.6}$$

It means that if $\mathbf{A}$ is a convex fuzzy set then its α-level set is a convex set for any $\alpha \in [0, 1]$.

A fuzzy number is a special kind of a fuzzy set on the universe of real numbers.

Definition 5 Let $\mathbf{A} \underset{\sim}{\subseteq} \mathcal{D}$ and $\mathcal{D} \subseteq \mathbb{R}$. $\mathbf{A}$ is called *fuzzy number* if it is convex and its kernel $\mathrm{Ker}(\mathbf{A}) = \{z_0\}$ where $z_0 \in \mathcal{D}$. To emphasize the kernel point we denote the fuzzy number by $\mathbf{Z_0}$.

Remark 2 *We should stress that the definition of a fuzzy number is not unique throughout the literature. For instance, some authors also require symmetry of the fuzzy set $\mathbf{Z_0}$ w.r.t. the node $z_0 \in \mathbb{R}$. On the other hand, there are several authors who admit to have a wider but closed kernel (i.e. a real interval). Such fuzzy set is often called "fuzzy interval" and the fuzzy number given by Definition 5 is then a special case of a fuzzy interval.*

1.1.2 T-norms

Original motivation for introducing the class of generalized multiplications known as *triangular norms* (t-norms) was not logical. The main idea was to generalize the concept of the triangular inequality. Since t-norms preserve the fundamental properties of the crisp conjunction, they become to interesting for fuzzy logic as its natural generalizations.

Definition 6 A binary operation $* : [0, 1]^2 \to [0, 1]$ is called *triangular norm* (t-norm) if it fulfills commutativity, associativity, monotonicity, and the following boundary condition holds i.e. for all $x, y, z \in [0,1]$:

$$\begin{aligned} x * y &= y * x && \text{(commutativity)}, \\ x * (y * z) &= (x * y) * z && \text{(associativity)}, \\ x \leq y &\Longrightarrow x * z \leq y * z && \text{(monotonicity)}, \\ x * 1 &= x && \text{(boundary condition)}. \end{aligned}$$

Example 1 *Below, we show the most known examples of continuous t-norms which serve as natural interpretations of a generalized conjunction:*

(1) *Minimum t-norm* $x * y = x \wedge y$,

(2) *Product t-norm* $x \odot y = x \cdot y$,

(3) *Łukasiewicz t-norm* $x \otimes y = \max(0, x + y - 1)$.

Another operation associated with the t-norm is called *triangular conorm* t-conorm and it corresponds (due to its behavior) to a generalization of the classical connective 'or'. It serves for also interpretation of unions of fuzzy sets.

Definition 7 A t-conorm is a binary operation $\sqcup : [0,1]^2 \to [0,1]$ which has the properties of commutativity, associativity and monotonicity (introduced in Definition 6) and fulfills the following boundary condition for all $x \in [0,1]$:

$$0 \sqcup x = x.$$

A t-conorm dual to a given t-norm $*$ is given by

$$a \sqcup b = 1 - (1 - a) * (1 - b).$$

Example 2 *The most important t-conorms dual to the t-norms from Example 1 are:*

(1) *Maximum t-conorm (dual to minimum t-norm)* $x \sqcup y = x \vee y$,

(2) *Product t-conorm (dual to product t-norm)* $x \sqcup y = x + y - x \cdot y$,

(3) *Łukasiewicz t-conorm (dual to Łukasiewicz t-norm)* $x \oplus y = \min(1, x + y)$.

Let us stress that *maximum* is the least t-conorm i.e. $x \vee y \leq x \sqcup y$ for all $x, y \in [0,1]$ and for any t-conorm $\sqcup$ (see [52]).

It follows from the definition of the t-norm that it is a monoidal operation on $[0,1]$. Furthermore, $\langle [0,1], \wedge, \vee \rangle$ is a complete lattice. Therefore, we can define the residuation operation in the following form.

Definition 8 Let $*$ be a t-norm. The *residuation* operation $\rightarrow_*\colon [0,1]^2 \rightarrow [0,1]$ is defined by

$$x \rightarrow_* y = \bigvee\{z \mid x * z \leq y\}. \tag{1.1.7}$$

Lemma 1 *Let $*$ be a left-continuous t-norm and $\rightarrow_*$ its residuation. Then the following properties hold for all $x, y, z \in [0,1]$:*

$$x * y \leq z \Longleftrightarrow y \leq x \rightarrow_* z, \tag{1.1.8}$$

$$x \leq y \Longrightarrow y \rightarrow_* z \leq x \rightarrow_* z, \tag{1.1.9}$$

$$x \leq y \Longrightarrow z \rightarrow_* x \leq z \rightarrow_* y. \tag{1.1.10}$$

The residuation operation serves as an operation representing the generalized implication for fuzzy logics.

Moreover, we will use the following derived operations

$$x^n = \underbrace{x * \ldots * x}_{n-times},$$

$$x \leftrightarrow_* y = (x \rightarrow_* y) \wedge (y \rightarrow_* x).$$

In the sequel, we denote Łukasiewicz operations t-norm, t-conorm and residuation by $\otimes$, $\oplus$ and $\rightarrow_\otimes$, respectively. It is worth to mention the following relation between Łukasiewicz t-conorm and residuation:

$$(1-x) \oplus y = x \rightarrow_\otimes y. \tag{1.1.11}$$

Let us recall some basic classes of t-norms.

Definition 9 Let $*$ be a t-norm. Then $*$ is called

- *Archimedean* if
$$\forall x, y \in (0,1)\ \exists n \in \mathbb{N}:\ x^n < y, \tag{1.1.12}$$

- *nilpotent* if
$$\forall x \in (0,1)\ \exists n \in \mathbb{N}:\ x^n = 0, \tag{1.1.13}$$

- *strict* if it is continuous and
$$\forall x, y, z \in [0,1], x > 0, y < z\ :\ x * y < x * z, \tag{1.1.14}$$

- *idempotent* if

$$\forall x \in [0,1] \ : \ x * x = x. \tag{1.1.15}$$

Let us stress that if $*$ is a continuous strict t-norm then there exists a continuous strictly increasing mapping $g : [0,1] \to [0,1]$ called *multiplicative generator* such that

$$x * y = g^{-1}(g(x) \cdot g(y)) \quad x, y, \in [0,1]. \tag{1.1.16}$$

1.1.3 Operations on Fuzzy Sets

Equality of fuzzy sets is defined as follows.

Definition 10 Let $\mathbf{A}, \mathbf{B} \underset{\sim}{\subset} \mathcal{D}$. Then $\mathbf{A} = \mathbf{B}$ if

$$\mathbf{A}(x) = \mathbf{B}(x) \tag{1.1.17}$$

for all $x \in \mathcal{D}$.

Definition 11 Let $\mathbf{A}, \mathbf{B} \underset{\sim}{\subset} \mathcal{D}$. Then the *union* $\mathbf{C} = \mathbf{A} \cup \mathbf{B}$ of these two fuzzy sets is a fuzzy set $\mathbf{C} \underset{\sim}{\subset} \mathcal{D}$ given as follows

$$\mathbf{C}(x) = \mathbf{A}(x) \vee \mathbf{B}(x). \tag{1.1.18}$$

Definition 12 Let $\mathbf{A}, \mathbf{B} \underset{\sim}{\subset} \mathcal{D}$. Then the *intersection* $\mathbf{C} = \mathbf{A} \cap \mathbf{B}$ of these two fuzzy sets is a fuzzy set $\mathbf{C} \underset{\sim}{\subset} \mathcal{D}$ given as follows

$$\mathbf{C}(x) = \mathbf{A}(x) \wedge \mathbf{B}(x). \tag{1.1.19}$$

Besides original Definitions 11 and 12, the $\sqcup$-union and $*$-intersection can be defined by t-conorms and t-norms [38], respectively. Given a t-norm $*$ and a t-conorm $\sqcup$, the respective $*$-intersection $\mathbf{C} = \mathbf{A} \cap_* \mathbf{B}$ is given by

$$\mathbf{C}(x) = \mathbf{A}(x) * \mathbf{B}(x) \tag{1.1.20}$$

and the respective $\sqcup$-union $\mathbf{C} = \mathbf{A} \cup_\sqcup \mathbf{B}$ is given by

$$\mathbf{C}(x) = \mathbf{A}(x) \sqcup \mathbf{B}(x). \tag{1.1.21}$$

1.1.4 Fuzzy Relations

Definition 13 Let $\mathbf{A} \underset{\sim}{\subset} \mathcal{D}$ and $\mathbf{B} \underset{\sim}{\subset} \mathcal{D}'$. Then their *fuzzy Cartesian product* $\mathbf{A} \times \mathbf{B} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ is a fuzzy set with the following membership function

$$(\mathbf{A} \times \mathbf{B})(x, y) = \mathbf{A}(x) \wedge \mathbf{B}(y) \tag{1.1.22}$$

for all $x \in \mathcal{D}$ and $y \in \mathcal{D}'$.

In formula (1.1.22), there is the minimum operation used but any t-norm could be used, in general [29, 48]. Then formula (1.1.22) is modified as follows

$$(\mathbf{A} \times_* \mathbf{B})(x, y) = \mathbf{A}(x) * \mathbf{B}(y). \tag{1.1.23}$$

For example, the product t-norm $x * y = x \odot y$ is very often used.

Definition 14 An *n-ary fuzzy relation* $\mathbf{R}$ is a fuzzy set on a Cartesian product $\mathcal{D}_1 \times \cdots \times \mathcal{D}_n$ of n universes.

The membership degree $\mathbf{R}(x_1, \ldots, x_n)$ expresses the degree, in which the n-tuple $(x_1, \ldots, x_n)$ is in the fuzzy relation $\mathbf{R}$. Obviously, classical relation is only a special type of fuzzy relation which maps the n-tuple either to 0 or to 1.

As well as in the classical mathematics, we can construct *compositions* of (fuzzy) relations. Sometimes, we use the notion *fuzzy relational products*. Most of them are motivated and studied in [3]. We will recall only two of them.

Definition 15 Let $\mathbf{R} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ and $\mathbf{S} \underset{\sim}{\subset} \mathcal{D}' \times \mathcal{D}''$ and let $*$ be a t-norm. The $\circ_*$ *composition* (also sup-$*$ composition) $\mathbf{R} \circ_* \mathbf{S} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}''$ of these two binary fuzzy relations is defined as follows

$$\mathbf{R} \circ_* \mathbf{S}(x, z) = \bigvee_{y \in \mathcal{D}'} (\mathbf{R}(x, y) * \mathbf{S}(y, z)), \quad x \in \mathcal{D}, z \in \mathcal{D}''. \tag{1.1.24}$$

Definition 16 Let $\mathbf{R} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ and $\mathbf{S} \underset{\sim}{\subset} \mathcal{D}' \times \mathcal{D}''$ and let $*$ be a left-continuous t-norm. The $\lhd_*$ *composition* (also inf$\rightarrow_*$ composition) $\mathbf{R} \lhd_* \mathbf{S} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}''$ of these two binary fuzzy relations is defined as follows

$$\mathbf{R} \lhd_* \mathbf{S}(x, z) = \bigwedge_{y \in \mathcal{D}'} (\mathbf{R}(x, y) \rightarrow_* \mathbf{S}(y, z)), \quad x \in \mathcal{D}, z \in \mathcal{D}'' \tag{1.1.25}$$

where $\rightarrow_*$ is the residuation operation adjoint to the t-norm $*$.

Remark 3 *The $\lhd_*$ composition which first appeared in [3] is sometimes called the Bandler-Kohout subproduct.*

Definitions of an *image of a fuzzy sets under a fuzzy relation* can be derived from the previous compositions of fuzzy relations.

Definition 17 Let $\mathbf{A} \underset{\sim}{\subset} \mathcal{D}$, $\mathbf{R} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ and let $*$ be a t-norm. Then the *direct image of* $\mathbf{A}$ *under* $\mathbf{R}$, $\mathbf{A} \circ_* \mathbf{R} \underset{\sim}{\subset} \mathcal{D}'$, is given as follows

$$\mathbf{A} \circ_* \mathbf{R}(y) = \bigvee_{x \in \mathcal{D}} (\mathbf{A}(x) * \mathbf{R}(x,y)), \quad y \in \mathcal{D}'. \tag{1.1.26}$$

Definition 18 Let $\mathbf{A} \underset{\sim}{\subset} \mathcal{D}$, $\mathbf{R} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ and let $*$ be a left-continuous t-norm. Then the *subdirect image of* $\mathbf{A}$ *under* $\mathbf{R}$, $\mathbf{A} \lhd_* \mathbf{R} \underset{\sim}{\subset} \mathcal{D}'$, is given as follows

$$\mathbf{A} \lhd_* \mathbf{R}(y) = \bigwedge_{x \in \mathcal{D}} (\mathbf{A}(x) \to_* \mathbf{R}(x,y)), \quad y \in \mathcal{D}'. \tag{1.1.27}$$

Let us stress that the Definitions 17 and 18 are very important because they constitute a mathematical basis for fuzzy inferences, see Section 1.2.

A special case arises when the fuzzy set $\mathbf{A}$ is a singleton i.e.

$$\mathbf{A}(x) = \begin{cases} 1, & x = x' \in \mathcal{D} \\ 0, & x \neq x' \end{cases} \tag{1.1.28}$$

Then the computation of the image defined by (1.1.27) of such a fuzzy set is simplified as follows

$$\mathbf{B}(y) = \mathbf{R}(x', y), \quad y \in \mathcal{D}' \tag{1.1.29}$$

no matter which composition is considered.

1.2 Fuzzy Modelling

Fuzzy modelling is a name for a very general field, say for methods of mathematical modelling using of the fuzzy set theory and the fuzzy logic. It focuses, especially, on the *fuzzy rule based systems* (FRB). These systems can be applied generally in many real problems, including the most often cited fuzzy control, decision-making and information retrieval.

1.2.1 FRB System Architecture

FRB system is a system consisting of four main blocks (*fuzzifier, fuzzy rule base, inference engine, defuzzifier*) typically of the structure displayed on Figure 1.1, see e.g. [8].

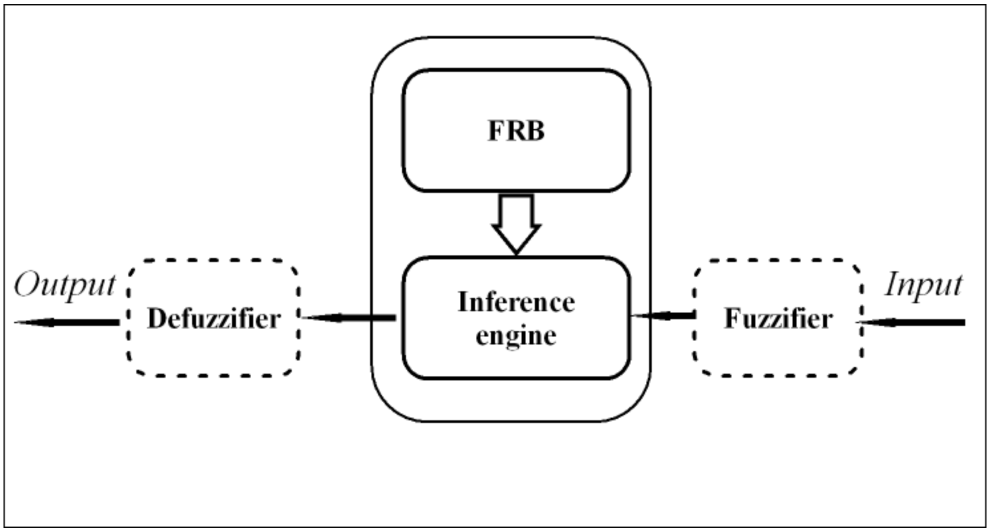


Figure 1.1: Architecture of an FRB system.

Fuzzifier is a block preforming a mapping from $\mathcal{D}$ to the set of all fuzzy sets on $\mathcal{D}$ (denoted by $\mathcal{F}(\mathcal{D})$). This mapping is involved since the inference engine can formally deal only with fuzzy inputs. The most usual one is a singleton fuzzifier which assigns a singleton (1.1.28) to each crisp input $x' \in \mathcal{D}$. Besides typical fuzzy control applications dealing only with measured crisp values, this block is not required in all FRB systems. Especially in particular decision-making problems when input values are hardly technically measurable, fuzzifiers are not implemented. Then only some rough evaluations given in a human language and modelled by fuzzy sets can be achieved.

Remark 4 *In more-dimensional models, we often meet such situations, when some inputs are connected to an inference engine directly and other are connected via fuzzifiers. For instance, in medical screening we expect crisp values determining a patient's temperature and linguistic expressions describing his/her subjective pain.*

Whole knowledge available in a given fuzzy rule base consisting of n fuzzy rules is contained in pairs of input-output fuzzy sets i.e. in $(\mathbf{A}_1, \mathbf{F}_1), \ldots, (\mathbf{A}_n, \mathbf{F}_n)$, telling us that, for $i = 1, ..., n$, the fuzzy set $\mathbf{A}_i \underset{\sim}{\subsetneq} \mathcal{D}$ is assigned the fuzzy set $\mathbf{F}_i \underset{\sim}{\subsetneq} \mathcal{D}'$ [78].

In general, there are two standard approaches to modelling a given fuzzy rule base by an appropriate fuzzy relation. Let $x \in \mathcal{D}$, $y \in \mathcal{D}'$, let operation $*$ be a left-continuous t-norm and finally, let $\rightarrow_*$ be its adjoint residuation operation. Then the first approach consists in a construction of a fuzzy relation $\hat{\mathbf{R}}_* \underset{\sim}{\subsetneq} \mathcal{D} \times \mathcal{D}'$ given as follows:

$$\hat{\mathbf{R}}_*(x, y) = \bigwedge_{i=1}^{n} (\mathbf{A}_i(x) \rightarrow_* \mathbf{F}_i(y)) \,. \tag{1.2.1}$$

As written in [20], "In the above view, each piece of information (fuzzy rule) is viewed as a constraint. This view naturally leads to a conjunctive way of merging the individual pieces of information since the more information, the more constraints and the less possible values to satisfy them."

This fact together with the fact that the minimum operation as well as other t-norms is an appropriate interpretation of a conjunction (logical connective AND) and residuation operations are appropriate interpretations of an implication [2, 18, 32, 52, 77] leads to a conclusion that $\hat{\mathbf{R}}_*$ is a proper model of the following fuzzy rules

$$\begin{array}{c} \textbf{IF } x \text{ is } \mathcal{A}_1 \quad \textbf{THEN} \quad y \text{ is } \mathcal{F}_1 \\ \cdots \\ \textbf{AND} \\ \cdots \\ \textbf{IF } x \text{ is } \mathcal{A}_n \quad \textbf{THEN} \quad y \text{ is } \mathcal{F}_n \end{array} \tag{1.2.2}$$

where $\mathcal{A}_i, \mathcal{F}_i$ are linguistic expression [51, 81] represented by fuzzy sets $\mathbf{A}_i \underset{\sim}{\subsetneq} \mathcal{D}$ and $\mathbf{F}_i \underset{\sim}{\subsetneq} \mathcal{D}'$, respectively.

The second standard approach to modelling a fuzzy rule base, which was initiated by a successful experimental application implemented by Mamdani and Assilian [42], consists in a construction of a fuzzy relation $\check{\mathbf{R}}_* \underset{\sim}{\subsetneq} \mathcal{D} \times \mathcal{D}'$ given as follows

$$\check{\mathbf{R}}_*(x, y) = \bigvee_{i=1}^{n} (\mathbf{A}_i(x) * \mathbf{F}_i(y)). \tag{1.2.3}$$

Obviously, fuzzy relation (1.2.3) can be hardly considered as a model of fuzzy rule base (1.2.2). As mentioned above, a t-norm operation is an appropriate interpretation of a conjunction, not implication and neither the maximum operation disjunctively aggregating all rules has anything common with the logical connective "AND".

We again recall the work of Dubois, Prade and Ughetto [20]. "It seem that fuzzy rules modelled by (1.2.3) are not viewed as constraints but are considered as pieces of data. Then the maximum in (1.2.3) expresses accumulation of data".

This fact together with a commonly known fact that the maximum operation as well as other t-conorms are appropriate interpretations of a disjunction (logical connective OR) [32, 52] leads to a conclusion that $\check{\mathbf{R}}_*$ given by (1.2.3) is a proper model of the following fuzzy rules

$$\begin{array}{ccc} x \text{ is } \mathcal{A}_1 & \textbf{AND} & y \text{ is } \mathcal{F}_1 \\ & \dots & \\ & \textbf{OR} & \\ & \dots & \\ x \text{ is } \mathcal{A}_n & \textbf{AND} & y \text{ is } \mathcal{F}_n. \end{array} \tag{1.2.4}$$

It is worth mentioning that distinguishing between the conditional (IF-THEN) form of fuzzy rules (1.2.2) and the Cartesian product (AND) form of fuzzy rules (1.2.4) on a syntactical level is not very common but it can be found e.g. in [11, 37, 49]. Usually only the form given by (1.2.2) is because of several (e.g. historical reasons or equivalence of both form sin the classical case) considered and the differences are taken into account only on a semantical level. But the differences can play a crucial role for further implementations and therefore they should be kept in mind. For more detailed study concerning both rule forms we refer to [32, 52, 37] and to an exhaustive investigation in [19].

Besides a fuzzy rule base, (fuzzy) inference mechanism is an essential part of each fuzzy rule based system depicted on Figure 1.1. It is a deduction rule determining an output $\mathbf{B} \underset{\sim}{\subset} \mathcal{D}'$ based on an arbitrary input $\mathbf{A} \underset{\sim}{\subset} \mathcal{D}$. Particularly, it is defined as an image of $\mathbf{A}$ under a fuzzy relation $\mathbf{R} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$, which models a given fuzzy rule base, see Section 1.1.

Defuzzifier is a block performing a mapping from the set of all fuzzy sets on $\mathcal{D}'$ (denoted by $\mathcal{F}(\mathcal{D}')$) to this domain $\mathcal{D}'$. This mapping is called *defuzzification* and it is especially required in automatized application such as fuzzy control when we finally have to give a precise control action (i.e. a number) to a controlled plant. This block can be replaced e.g. by a linguistic approximation block which finds the closest fuzzy set from some predefined finite set of fuzzy sets from $\mathcal{F}(\mathcal{D}')$ to a fuzzy output $\mathbf{B} \underset{\sim}{\subset} \mathcal{D}'$. Usually, the predefined finite set consists of fuzzy sets modelling evaluating linguistic expressions with clear linguistic meanings and therefore the final answer is the respective label of the closest fuzzy set - *a word*.

Let us mention some of the defuzzification methods. The *Center of Gravity* (COG) method:

$$\mathrm{COG}(\mathbf{B}) = \frac{\int_{\mathcal{D}'} y \cdot \mathbf{B}(y)dy}{\int_{\mathcal{D}'} \mathbf{B}(y)dy} \tag{1.2.5}$$

which is usually in practice simplified by discrete sums applied instead of the integrals.

Another useful defuzzification is the *Mean of Maxima* (MOM) method:

$$\mathrm{MOM}(\mathbf{B}) = \frac{1}{n_{max}} \sum_{j=1}^{n_{max}} y_j^{max} \tag{1.2.6}$$

where $y_j^{max} = y_j$ if $\mathbf{B}(y_j) = \max\{\mathbf{B}(y) \mid y \in \mathcal{D}'\}$. It means that the method locates an arithmetic mean of the set $Y = \{y_j^{max} \mid 1, \ldots, n_{max}\}$ of all elements of the support $\mathrm{Supp}(\mathbf{B})$ with maximal membership degree. In the case of uncountable set Y, the defuzzification is computed as an arithmetic mean of $\sup Y$ and $\inf Y$.

Defuzzification *First of Maxima* (FOM) chooses the first element of $\mathrm{Supp}(\mathbf{B})$ with maximal membership degree:

$$\mathrm{FOM}(\mathbf{B}) = \inf Y \tag{1.2.7}$$

and Defuzzification *Last of Maxima* (LOM) analogously chooses the last element of $\mathrm{Supp}(\mathbf{B})$ with maximal membership degree:

$$\mathrm{LOM}(\mathbf{B}) = \sup Y. \tag{1.2.8}$$

Defuzzifications FOM and LOM are not usually directly implemented but their significant importance is given by the fact that they are used in the construction of the Defuzzification of Evaluating Expressions (DEE), see [48, 50]. This method classifies the output fuzzy set $\mathbf{B}$ into three classes (small, medium, big), at first, and then defuzzifies by FOM, MOM and LOM w.r.t. the fuzzy set class, respectively. Results of the DEE defuzzification are depicted on Figure 1.2.

1.2.2 Fuzzy Modelling on the Basis of Fuzzy Interpolation

Besides typical fuzzy control implementations with crisp inputs, fuzzifiers and defuzzifiers, we have to keep in mind also applications without fuzzifiers and with direct fuzzy inputs as discussed above. An FRB defines a partial mapping from $\mathcal{F}(\mathcal{D})$ to $\mathcal{F}(\mathcal{D}')$ such that it assigns $\mathbf{F}_i \underset{\sim}{\subset} \mathcal{D}'$ to $\mathbf{A}_i \underset{\sim}{\subset} \mathcal{D}$ for any $i = 1, \ldots, n$ and

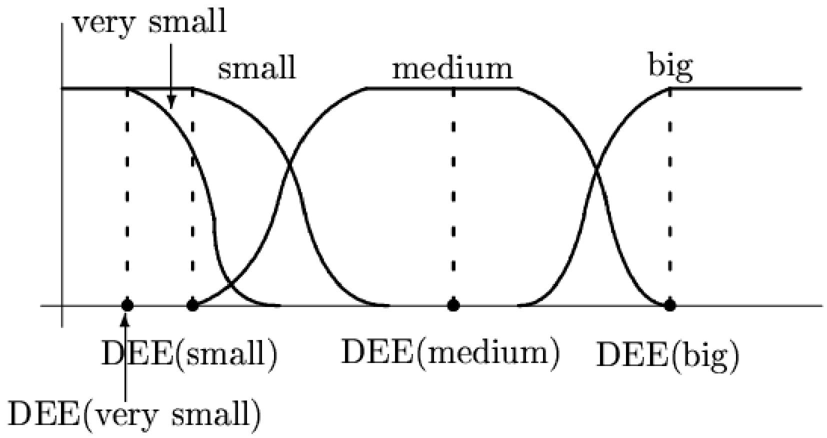


Figure 1.2: Some fuzzy sets modelling evaluating linguistic expressions. Defuzzification of evaluating linguistic expressions (DEE) is depicted on the domain.

the corresponding inference engine extends the mapping to a total one. To keep the partial mapping defined by the given FRB even after the extension to a total mapping is an obvious requirement which leads to the fundamental fuzzy interpolation condition:

$$\mathbf{A}_i \Box_* \mathbf{R} = \mathbf{F}_i, \quad i = 1, \ldots, n \tag{1.2.9}$$

where $\Box_*$ is an image determined by the inference mechanism and $\mathbf{R} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ is a fuzzy relation interpreting the FRB. Keeping condition (1.2.9) certifies good behaviour of an FRB system from the interpolation point of view and therefore it should be always kept in mind throughout an identification of a model when choosing inference mechanism and an appropriate FRB interpretation.

Remark 5 *Interpolation condition (1.2.9) leads to the so called systems of fuzzy relation equation which have been studied by many authors, see [15, 30, 35]. Besides the mentioned choices through the identification process they also answer the solvability condition questions i.e. finding the conditions to antecedent fuzzy sets* $\mathbf{A}_i$ *and consequent fuzzy sets* $\mathbf{F}_i$ *under which some interpolating extension exists.*

1.2.3 Takagi-Sugeno Rules

T. Takagi and M. Sugeno in [76] introduced rules with consequents being equal to linear expressions:

$$\textbf{IF } x \text{ is } \mathcal{A}_i \textbf{ THEN } y \text{ is } a_i + b_i x \quad i = 1, \ldots, n \tag{1.2.10}$$

where a_i, b_i are real numbers.

Takagi-Sugeno FRB system explicitly works with crisp inputs and the inference is performed according to the following formula

$$f^A(x) = \frac{\sum_{i=1}^{n} \mathbf{A}_i(x) \cdot (a_i + b_i x)}{\sum_{i=1}^{n} \mathbf{A}_i(x)} \tag{1.2.11}$$

which means that we get a real-valued function $f^A(x)$ which usually approximates some dependency and no defuzzifier is required anymore.

Obviously, Takagi-Sugeno rules can be directly generalized to any real valued function in the consequent parts of the rules. Usually, polynomials $p_i^n(x)$ of the n-th order are considered. On the other hand, the inference function given by (1.2.11) is due to the weighting influence of the antecedent fuzzy sets $\mathbf{A}_i$ already non-linear even with linear consequents $p_i^1(x)$. Therefore polynomials up to the first order are supposed to be enough.

1.3 Fuzzy Modelling on the Basis of Fuzzy Approximation

Fuzzy approximation is understood as a collection of techniques for an approximation of functional dependencies between variables by means of the fuzzy set theory or fuzzy logic [9, 32, 52]. It is a quickly developing mathematical branch aiming at approximation of some dependencies by means of the fuzzy set theory and the fuzzy logic in broader sense. We can find its roots in the Takagi-Sugeno fuzzy rule based systems [76] and in works aiming at approximation capabilities of fuzzy rule based systems, see [6, 40]. Nowadays, there are two main approaches to the fuzzy approximation.

1.3.1 Normal Forms

The first particular approach relates to the so called *normal forms* [13, 56] consists in an investigation of different interpretations of fuzzy IF-THEN rules. Typically, there are two standard interpretations, namely disjunction of conjunctions and conjunction of implications [21, 46, 49]. A variety of interpretations of logical connectives leads to different formalizations of fuzzy rules.

The crucial notion in the field of normal forms is the *similarity relation* sometimes also called *fuzzy equality* which is a binary fuzzy relation naturally generalizing the classical equality. Such a similarity relation is then a relation $\mathbf{S} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}$ which is reflexive, symmetric and transitive w.r.t a given t-norm, for more details see Chapter 5.

Let us be given a fuzzy set $\mathbf{F} \underset{\sim}{\subset} \mathcal{D}$, a left-continuous t-norm $*$ and a similarity relation $\mathbf{S} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}$ and let $\mathcal{N} \subseteq \mathcal{D}$. Then the disjunctive and the conjunctive normal forms of $\mathbf{F}$ [59] are defined as follows

$$\mathbf{F}_{DNF,*}(x) = \bigvee_{c \in \mathcal{N}} (\mathbf{S}(c, x) * \mathbf{F}(c)) \tag{1.3.1}$$

$$\mathbf{F}_{CNF,*}(x) = \bigwedge_{c \in \mathcal{N}} (\mathbf{S}(x, c) \rightarrow_* \mathbf{F}(c)), \tag{1.3.2}$$

respectively.

These normal forms of $\mathbf{F}$ approximate the fuzzy set $\mathbf{F}$ and their approximation abilities are studied. In the extreme case, when $\mathcal{N} = \mathcal{D}$ the normal forms of $\mathbf{F}$ are precisely equal to $\mathbf{F}$. For details we again refer to Chapter 5.

A natural question, what is the reason to approximate a fuzzy set, can appear. A fuzzy rule base is a rough description of some, usually functional, dependency. And range $\mathcal{D}'$ of each continuous function f on a compact domain $\mathcal{D}$ is a closed interval which can be transformed by an appropriate mapping to $[0, 1]$. So, from $f : \mathcal{D} \rightarrow \mathcal{D}'$ we obtain a fuzzy set $\mathbf{F} \underset{\sim}{\subset} \mathcal{D}$. Therefore by an approximation of fuzzy sets where we can use operations serving as interpretations of logical connectives such as t-norms and residuations we consequently approximate continuous functions.

1.3.2 Fuzzy Transform

The second approach uses the fuzzy set theory to approximate a given function by another one preserving the feature of transparent interpretability. The main idea is to characterize output values w.r.t. local domains and then to aggregate them into the global information. The *fuzzy transform* technique [54, 57, 58] follows this idea and expresses a functional dependency as a linear combination of basic functions, see Chapter 2. Note that a function produced by this technique may be treated as a basis for singleton models e.g. Takagi-Sugeno models of the 0-th order .

1.4 Neural Networks

The neural network area is a field which investigates architectures of *neurons* (units) and *dendrites* (connections). These architectures have been motivated by the neural structure of a human brain as easily seen from the neuron structure, displayed on Figure 1.3,

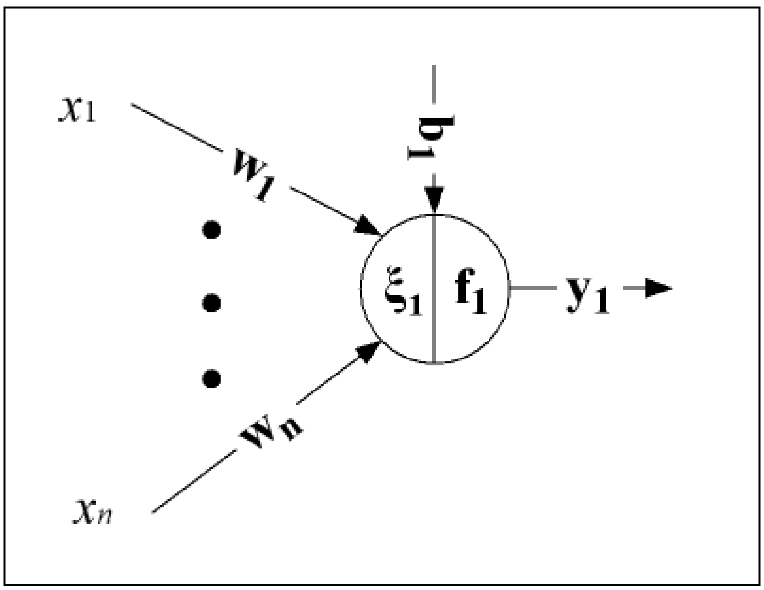


Figure 1.3: Neuron scheme.

One neuron, as a computational unit, deals with $n \in \mathbb{N}$ real inputs $x_1, \ldots, x_n$ which are weighted by parameters $w_1, \ldots, w_n$ and a bias $b_1 \in \mathbb{R}$. There is a synaptical operation $\xi_1 : \mathbb{R}^{(n+1)} \times \mathbb{R}^n$ on the synapses of the dendrites and the unit which usually computes the weighted sum inputs subtracted by the bias value b_1:

$$\xi_1(x_1, \ldots, x_n, w_1, \ldots, w_n, b_1) = \sum_{i=1}^{n} w_i \cdot x_i - b_1 \tag{1.4.1}$$

so, the bias plays the role of a threshold. The real value computed according to (1.4.1) is then an argument of *activation function* $f_1 : \mathbb{R} \to \mathbb{R}$ which gives a final output of the neuron. This output is either a final output from the network it is an input to another neuron.

The main feature of the neural networks is their ability to *learn* which means to tune their parameters to minimize some error function [26, 41, 66].

Methods from both soft computing areas - fuzzy approximation and neural networks - have been proven to be universal approximators and both preserve the specific branch advantages (learning vs. interpretability). Therefore they merit a comparison and a joint study, see Chapter 8.

Chapter 2

Fuzzy Transform

This chapter overviews main definitions and ideas of the F-transform technique - a concrete method of the fuzzy approximation of a continuous function. This technique was firstly introduced by I. Perfilieva in [53] and then recalled with extending results, developments and applications (see e.g. [54, 57, 58]).

2.1 Fuzzy Partition - Basic Functions

Let us denote the domain of all functions considered in this chapter by $\mathcal{D} = [a, b]$. The domain is then partitioned by a *fuzzy partition* (see [62]) i.e. by fuzzy sets $\mathbf{A}_i \underset{\sim}{\subseteq} \mathcal{D}$, $i = 1, \ldots, n$ which fulfil the following Ruspini [62] condition:

$$\sum_{i=1}^{n} \mathbf{A}_i(x) = 1, \tag{2.1.1}$$

for all $x \in \mathcal{D}$.

Basic functions defined in [53] are special fuzzy sets forming a fuzzy partition. There are special assumptions required on the basic functions which yield appropriate properties which can be very valuable in further numerical methods.

Definition 19 Let $c_0 = c_1 < \cdots < c_n = c_{n+1}$ be fixed nodes within $[a, b]$ such that

$c_1 = a$, $c_n = b$ and $n \geq 2$. We say that fuzzy sets $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are *basic functions* and form a *fuzzy partition* of $\mathcal{D}$ if the following conditions hold true for each $i = 1, \ldots, n$:

1. $\mathbf{A}_i : [a, b] \to [0, 1],\ \mathbf{A}_i(c_i) = 1$,
2. $\mathbf{A}_i(x) = 0$ if $x \notin (c_{i-1}, c_{i+1})$;
3. $\mathbf{A}_i$ is continuous on $\mathcal{D}$;
4. $\mathbf{A}_i$ strictly increases on $[c_{i-1}, c_i]$ and strictly decreases on $[c_i, c_{i+1}]$;
5. $\sum_{i=1}^{n} \mathbf{A}_i(x) = 1$, for all $x \in X$.

If the nodes $c_1, \ldots, c_n$ are equidistant i.e. $c_i = a + h(i-1), i = 1, \ldots, n$ where $h = (b-a)/(n-1)$ and the following two additional properties are met:

6 $\mathbf{A}_i(c_i - x) = \mathbf{A}_i(c_i + x)$, for all $x \in [0, h],\ i = 2, \ldots, n-1,\ n > 2$,

7 $\mathbf{A}_{i+1}(x) = \mathbf{A}_i(x - h)$, for all $x \in [a + h, b],\ i = 2, \ldots, n-2,\ n > 2$,

we call the fuzzy partition *uniform* and talk about *uniform basic functions*indexuniform basic functions.

Remark 6 *In some cases, a general fuzzy partition works better than the uniform one. This fact raises a question how to construct basic functions in general. This can be done by some fuzzy clustering method (e.g.* c-means *[4]) when the basic functions will be the resulting clusters. Such approach can be helpful but uniform basic functions keep some nice and useful properties and therefore an approximation of functions using uniform basic functions as well. But it is worth mentioning that for main properties like a convergence this uniformity is not necessary. Since the proofs are analogous they will be omitted and everything will be proved only for uniform fuzzy partitions.*

For example, triangular shaped basic functions (see Fig. 2.1(a)) are given by

$$\mathbf{A}_i(x) = \begin{cases} \frac{(x-c_{i-1})}{c_i-c_{i-1}} & x \in [c_{i-1}, c_i] \\ \frac{(c_{i+1}-x)}{c_{i+1}-c_i} & x \in [c_i, c_{i+1}] \\ 0 & \text{otherwise} \end{cases} \tag{2.1.2}$$

where $i = 0, \ldots, n+1$ and $c_0 = c_1,\ c_{n+1} = c_n$, while sinusoidal shaped basic functions (see Fig. 2.1(b)) are given by

$$\mathbf{A}_i(x) = \begin{cases} \frac{1}{2}\left(\cos\left(\frac{\Pi(x-c_i)}{(c_i-c_{i-1})}\right)+1\right) & x \in [c_{i-1}, c_i] \\ \frac{1}{2}\left(\cos\left(\frac{\Pi(x-c_i)}{(c_{i+1}-c_i)}\right)+1\right) & x \in [c_i, c_{i+1}] \\ 0 & \text{otherwise} \end{cases} \tag{2.1.3}$$

where $i = 0, \ldots, n+1$ and $c_0 = c_1,\ c_{n+1} = c_n$.

First, let us recall the following lemma proved in [53] confirming that the definite integral of a basic function from a uniform fuzzy partition does not depend on its shape.

Lemma 2 *Let a uniform fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n$, $n \geq 2$. Then*

$$\int_a^b \mathbf{A}_1(x)dx = \int_a^b \mathbf{A}_n(x)dx = \frac{h}{2} \tag{2.1.4}$$

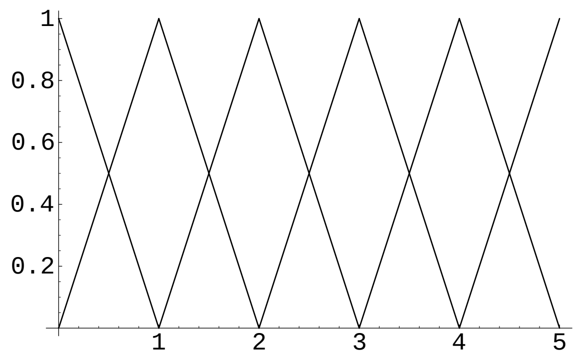
and for $i = 2, \ldots, n-1$

$$\int_a^b \mathbf{A}_i(x)dx = h. \tag{2.1.5}$$

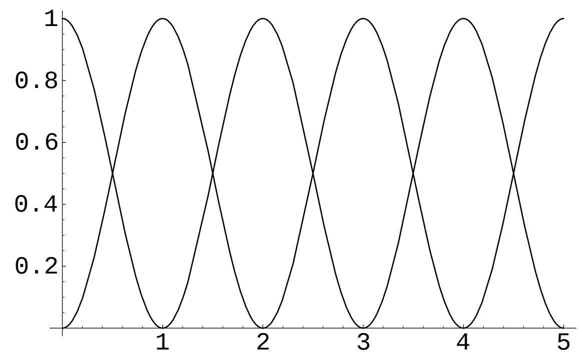
Let us create a fuzzy partition which is weaker than the uniform one.

Definition 20 Let $c_0 = c_1 < \cdots < c_n = c_{n+1}$ be fixed nodes within $\mathcal{D}$ such that $c_1 = a$, $c_n = b$ and $n \geq 2$ and let fuzzy sets $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are basic functions. Denote $h_i = c_{i+1} - c_i$.We say that the basic functions determine a *fuzzy partition with a symmetry* if the following condition

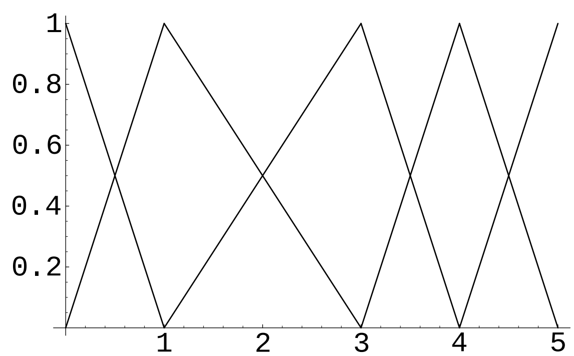
$$\mathbf{A}_i(c_i + x) = \mathbf{A}_{i+1}(c_{i+1} - x),\ x \in [0, h_i] \tag{2.1.6}$$



(a) Triangular shaped uniform fuzzy partition



(b) Sinusoidal shaped uniform fuzzy partition



(c) Triangular shaped non-uniform fuzzy partition with a symmetry

Figure 2.1: Graphic presentation of distinct fuzzy partitions.

holds for $i = 1, \dots, n-1$ then

Now, we prove that a fuzzy partition with a symmetry fulfils a similar property to the one from Lemma 2.

Lemma 3 *Let a fuzzy partition with a symmetry of $\mathcal{D}$ be given by basic functions* $\mathbf{A}_1, \dots, \mathbf{A}_n \subsetneq \mathcal{D}, n > 2$. *Then*

$$\int_a^b \mathbf{A}_i(x)dx = \frac{(h_{i-1} + h_i)}{2}. \tag{2.1.7}$$

PROOF: Let us express the value h_i with help of integrals:

$$h_i = \int_{c_i}^{c_{i+1}} 1dx = \int_{c_i}^{c_{i+1}} (\mathbf{A}_i(x) + \mathbf{A}_{i+1}(x))dx = \int_{c_i}^{c_{i+1}} \mathbf{A}_i(x)dx + \int_{c_i}^{c_{i+1}} \mathbf{A}_{i+1}(x)dx. \tag{2.1.8}$$

From (2.1.6) we get

$$\int_{c_i}^{c_{i+1}} \mathbf{A}_i(x)dx = \int_{c_i}^{c_{i+1}} \mathbf{A}_{i+1}(x)dx \tag{2.1.9}$$

and by (2.1.8), both sides of equation (2.1.9) are equal to $h_i/2$. Equation (2.1.7) which proves the lemma. □

It is easy to see that Lemma 3 generalizes Lemma 2 because conditions of uniformity of a fuzzy partition imply property (2.1.6) and obviously $h_0 = h_n = 0$. Let us consider the right hand side of (2.1.6). Then due to properties of uniform fuzzy partition we can write

$$\mathbf{A}_{i+1}(c_{i+1} - x) = \mathbf{A}_i(c_{i+1} - x - h) = \mathbf{A}_i(c_i - x) = \mathbf{A}_i(c_i + x)$$

which means that property (2.1.6) is fulfilled. Example is on Figure 2.1(c)

2.2 Fuzzy Transform

2.2.1 Direct Fuzzy Transform

Definition 21 [53] Let a fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \subsetneqq \mathcal{D}, n > 2$ and let $f : \mathcal{D} \to \mathbb{R}$ be an arbitrary function from $C(\mathcal{D})$. The n-tuple of real numbers $[F_1, \ldots, F_n]$ given by

$$F_i = \frac{\int_a^b f(x)\mathbf{A}_i(x)dx}{\int_a^b \mathbf{A}_i(x)dx}, \ i = 1, \ldots, n \tag{2.2.1}$$

is the *direct fuzzy transform (F-transform)* of f with respect to the given fuzzy partition. $F_1, \ldots, F_n$ are the *components* of the F-transform of f.

By Lemma 2, the direct F-transform of a function f can be simplified in the case of a uniform fuzzy partition as follows

$$F_1 = \frac{2}{h} \int\limits_{c_1}^{c_2} f(x)\mathbf{A}_1(x)dx \tag{2.2.2}$$

$$F_n = \frac{2}{h} \int\limits_{c_{n-1}}^{c_n} f(x)\mathbf{A}_n(x)dx \tag{2.2.3}$$

$$F_i = \frac{1}{h} \int\limits_{c_{i-1}}^{c_{i+1}} f(x)\mathbf{A}_i(x)dx, \quad i = 2, \ldots, n-1. \tag{2.2.4}$$

By Lemma 3, we can simplify in the case of a fuzzy partition with a symmetry the F-transform as follows

$$F_i = \frac{2}{(h_{i-1} + h_i)} \int_a^b f(x)\mathbf{A}_i(x)dx. \tag{2.2.5}$$

If we are not given an analytical description of a function f, but we are provided with function values at some nodes, say $p_1, \ldots, p_N$, then the discrete F-transform can be defined.

Definition 22 [54] Let a fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \subsetneqq \mathcal{D}, n > 2$ and let $f : \mathcal{D} \to \mathbb{R}$ be a function known at nodes $p_1, \ldots, p_N$ such that for each $i = 1, \ldots, n$, there exists $k = 1, \ldots, N$: $\mathbf{A}_i(p_k) > 0$. The n-tuple of reals $[F_1, \ldots, F_n]$ given by

$$F_i = \frac{\sum_{k=1}^{N} f(p_k)\mathbf{A}_i(p_k)}{\sum_{k=1}^{N} \mathbf{A}_i(p_k)}, \ i = 1, \ldots, n \tag{2.2.6}$$

is the *discrete direct F-transform* of f with respect to the given fuzzy partition.

Remark 7 *Throughout the text, if possible, we will omit the word "discrete" and use just the F-transform notion since from the context it is usually obvious whether we mean the one given by (2.2.6) or the one given by(2.2.1).*

Due to the usage of definite integrals in formula (2.2.1) and summations in formula (2.2.6) the linearity of the direct F-transform is an expected property. It can be formalized as follows.

Lemma 4 *[54] Let a fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \subsetneqq \mathcal{D}, n > 2$ and let f, g, h be continuous functions on $\mathcal{D}$ such that $h = \alpha f + \beta g$ where α, β are real numbers. Then the following equality holds*

$$[H_1, \ldots, H_n] = \alpha[F_1 \ldots, F_n] + \beta[G_1, \ldots, G_n] \tag{2.2.7}$$

where $[H_1, \ldots, H_n], [F_1, \ldots, F_n]$ and $[G_1, \ldots, G_n]$ are the F-transforms of h, f and g with respect to the given fuzzy partition, respectively .

The following lemma has been firstly published in [53] and the proof can be found in [54]. In fact, the lemma confirms the closeness of the components of the direct F-transform to the precise values of a given function at nodes c_i.

Lemma 5 *[53] Let a uniform fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \underset{\sim}{\subset} \mathcal{D}, n > 2$ and let $f \in C(\mathcal{D})$ be a twice continuously differentiable function on (a, b). Then for each $i = 1, \ldots, n$*

$$F_i = f(c_i) + O(h^2). \tag{2.2.8}$$

Lemma 3 enables us to generalize Lemma 5.

Lemma 6 *Let a fuzzy partition with a symmetry of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \underset{\sim}{\subset} \mathcal{D}, n > 2$ and let $f \in C(\mathcal{D})$ be a twice continuously differentiable function on (a, b). Then for each $i = 1, \ldots, n$*

$$F_i = f(c_i) + O(\max(h_{i-1}^2, h_i^2)). \tag{2.2.9}$$

PROOF: The proof uses an analogous technique to the one used in the proof of Lemma 5, see [54]. For arbitrary $i = 2, \ldots, n-1$:

$$F_i = \frac{2}{(h_{i-1} + h_i)} \left(\int_{c_{i-1}}^{c_i} f(x)\mathbf{A}_i(x)dx + \int_{c_i}^{c_{i+1}} f(x)\mathbf{A}_i(x)dx \right) =$$
$$\frac{2}{(h_{i-1} + h_i)} \left[\frac{h_{i-1}}{2} (f(c_{i-1})\mathbf{A}_i(c_{i-1}) + f(c_i)\mathbf{A}_i(c_i)) + O(h_{i-1}^3) \right] +$$
$$\frac{2}{(h_{i-1} + h_i)} \left[\frac{h_i}{2} (f(c_i)\mathbf{A}_i(c_i) + f(c_{i+1})\mathbf{A}_i(c_{i+1})) + O(h_i^3) \right] =$$
$$\frac{h_{i-1}}{(h_{i-1} + h_i)} f(c_i) + \frac{h_i}{(h_{i-1} + h_i)} f(c_i) + 2\frac{O(h_{i-1}^3) + O(h_i^3)}{(h_{i-1} + h_i)} =$$
$$f(c_i) + \frac{O(h_{i-1}^3 + h_i^3)}{(h_{i-1} + h_i)}$$

which equals to $f(c_i) + O(\max(h_{i-1}^2, h_i^2))$.

For the two remaining cases we can write

$$F_1 = \frac{2}{h_1} \left(\int_{c_1}^{c_2} f(x)\mathbf{A}_1(x)dx \right) =$$

$$\frac{2}{h_1}\left[\frac{h_1}{2}(f(c_1)\mathbf{A}_1(c_1) + f(c_2)\mathbf{A}_1(c_2)) + O(h_1^3)\right] = f(c_1) + O(h_1^2) \quad \text{and}$$

$$F_n = \frac{2}{h_n}\left(\int_{c_{n-1}}^{c_n} f(x)\mathbf{A}_n(x)dx\right) =$$

$$\frac{2}{h_n}\left[\frac{h_n}{2}(f(c_{n-1})\mathbf{A}_n(c_{n-1}) + f(c_n)\mathbf{A}_n(c_n)) + O(h_{n-1}^3)\right] =$$

$$f(c_n) + O(h_{n-1}^2)$$

which since $h_0 = c_1 - c_0 = a - a = 0$ and $h_n = c_{n+1} - c_n = b - b = 0$ coincides with formula (2.2.9). □

Lemma 7 *[53] Let a uniform fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \underset{\sim}{\subsetneq} \mathcal{D}, n > 2$ and let $f \in C(\mathcal{D})$ be a twice continuously differentiable function on (a, b). Then for each $i = 2, \ldots, n-1$*

$$\int_a^{c_i} f(x)dx = h\left(\frac{1}{2}F_1 + F_2 + \cdots + F_{i-1} + \frac{1}{2}F_i\right) + O(h^2). \tag{2.2.10}$$

Moreover, the integral $\int_a^b f(x)dx$ can be computed precisely:

$$\int_a^b f(x)dx = h\left(\frac{1}{2}F_1 + F_2 + \cdots + F_{n-1} + \frac{1}{2}F_n\right). \tag{2.2.11}$$

Lemma 7 clarifies the relationship between the computation of definite integrals and the F-transform. It shows how the F-transform can be used in numerical computations. In practical situations, the F-transform components are computed according to formula (2.2.6) of the discrete F-transform and formula (2.2.11) provides us with a numerical integral of f. Due to Lemma 3, we can present the following generalization valid eve for non-uniform fuzzy partitions.

Lemma 8 *Let a fuzzy partition with a symmetry of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \underset{\sim}{\subsetneq} \mathcal{D}, n > 2$ and let $f \in C(\mathcal{D})$ be a twice continuously differentiable* *function on (a,b). Then for each $i = 2, \dots, n-1$*

$$\int_a^{c_i} f(x)dx = \left(\frac{h_1}{2}F_1 + \frac{(h_1+h_2)}{2}F_2 + \cdots + \frac{(h_{i-2}+h_{i-1})}{2}F_{i-1}\frac{h_{i-1}}{2}F_i \right) + O(\max(h_{i-1}^2, h_i^2)). \tag{2.2.12}$$

Moreover, the integral $\int_a^b f(x)dx$ can be computed precisely

$$\int_a^b f(x)dx = \frac{1}{2}\left(\sum_{i=1}^{n-1} h_i F_i + \sum_{i=2}^{n} h_{i-1}F_i \right). \tag{2.2.13}$$

PROOF:

$$\int_a^{c_i} f(x)dx = \int_a^{c_i} f(x)\sum_{j=1}^{i} f(x)\mathbf{A}_j(x)dx = \int_{c_1}^{c_2} f(x)\mathbf{A}_1(x)dx +$$

$$\int_{c_1}^{c_3} f(x)\mathbf{A}_2(x)dx + \cdots + \int_{c_{i-2}}^{c_i} f(x)\mathbf{A}_{i-1}(x)dx + \int_{c_{i-1}}^{c_i} f(x)\mathbf{A}_i(x)dx$$

which due to Lemma 3 and using trapezium formula equals to

$$\frac{h_1}{2}F_1 + \frac{(h_1+h_2)}{2}F_2 + \cdots + \frac{(h_{i-2}+h_{i-1})}{2}F_{i-1} + \frac{h_{i-1}}{2}(f(c_{i-1})\mathbf{A}_i(c_{i-1}) + f(c_i)\mathbf{A}_i(c_i)) + O(h_{i-1}^2) \tag{2.2.14}$$

and because of Lemma 6 we get the proof of the first part of the lemma.

And analogously

$$\int_a^b f(x)dx = \int_a^b f(x)\sum_{i=1}^{n} f(x)\mathbf{A}_i(x)dx = \sum_{i=1}^{n}\int_a^b f(x)\mathbf{A}_i(x)dx =$$

$$\frac{h_1}{2}F_1 + \frac{(h_1+h_2)}{2} + \cdots + \frac{(h_{n-2}+h_{n-1})}{2}F_{n-1} + \frac{h_{n-1}}{2}F_n$$

which proves the second part.

□

If we deal with an approximation of a function it is necessary to distinguish the approximation among other possible ones. This is usually guaranteed by a minimization of a certain criterion which defines a closeness of an original function and its approximation in some natural sense.

In [54], there was the following criterion

$$\Phi(Q_1, \dots, Q_n) = \int_a^b \left(\sum_{i=1}^{n} (f(x) - Q_i)^2 \mathbf{A}_i(x) \right) dx \tag{2.2.15}$$

proposed to measure the quality of a discrete approximation of a function. By direct computation, one can check that the error function $\Phi : \mathbb{R}^n \to \mathbb{R}^+$ called *piecewise integral least square criterion*[54] is minimized by the components of the F-transform F_i defined by (2.2.6).

In the case of the discrete F-transform, the criterion is modified to the following one

$$\Phi(Q_1, \dots, Q_n) = \sum_{k=1}^{N} \left(\sum_{i=1}^{n} (f(p_k) - Q_i)^2 \mathbf{A}_i(p_k) \right). \tag{2.2.16}$$

2.2.2 Inversion

The direct F-transform of an original function serves as its discrete representation which can be e.g. successfully used in numerical computations. To bring the direct F-transform back we use the inverse F-transform; see [53].

Definition 23 Let $[F_1, \dots, F_n]$ be the direct F-transform of f with respect to $\mathbf{A}_1, \dots, \mathbf{A}_n \underset{\sim}{\subset} \mathcal{D}$. Then the function

$$f_n^F(x) = \sum_{i=1}^{n} F_i \mathbf{A}_i(x), \tag{2.2.17}$$

is called the *inverse F-transform* of f.

Let us fix a fuzzy partition. Then the inverse F-transform is a mapping which maps an n-tuple of reals to the space $lin_n\{\mathbf{A}_i\}$ of linear combinations of the basic functions.

Lemma 9 *[78] Let a fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \subsetneqq \mathcal{D}, n > 2$. Then each function $q \in lin_n\{\mathbf{A}_i\}$ i.e. $q(x) = \sum_{i=1}^n Q_i\mathbf{A}_i(x)$ is uniquely determined by the n-tuple $[Q_1, \ldots, Q_n]$.*

PROOF: By contrary, suppose that

$$\sum_{i=1}^{n} Q_i\mathbf{A}_i(x) = \sum_{i=1}^{n} R_i\mathbf{A}_i(x)$$

for $Q_1, \ldots, Q_n, R_1, \ldots, R_n \in \mathbb{R}$ such that there exists i: $Q_i \neq R_i$. For that index i, $\mathbf{A}_i(c_i) = 1$, whence $\mathbf{A}_j(c_i) = 0$ for $j \neq i$. This implies $Q_i = R_i$ which contradicts the assumption. $\square$

By Lemma 9, the space $lin_n\{\mathbf{A}_i\}$ is in a one-to-one correspondence with the set of n-tuples of reals, that is, with the set $\mathbb{R}^n$.

The inverse F-transform gives us a continuous function. A sequence of the inverse F-transforms uniformly converges to the original function f. This fact has been formulated and proved in the following theorem and its two corollaries, see [58].

Theorem 2 *[58] Let f be a continuous functions on $\mathcal{D}$. Then for any $\varepsilon > 0$ there exists $n_\varepsilon > 2$ and a fuzzy partition $\mathbf{A}_1, \ldots, \mathbf{A}_{n_\varepsilon}$ of $\mathcal{D}$ such that for all $x \in X$*

$$|f(x) - f^F_{n_\varepsilon}(x)| \leq \varepsilon, \tag{2.2.18}$$

where $f^F_{n_\varepsilon}$ is the inverse F-transform of f with respect to the fuzzy partition $\mathbf{A}_1, \ldots, \mathbf{A}_{n_\varepsilon}$.

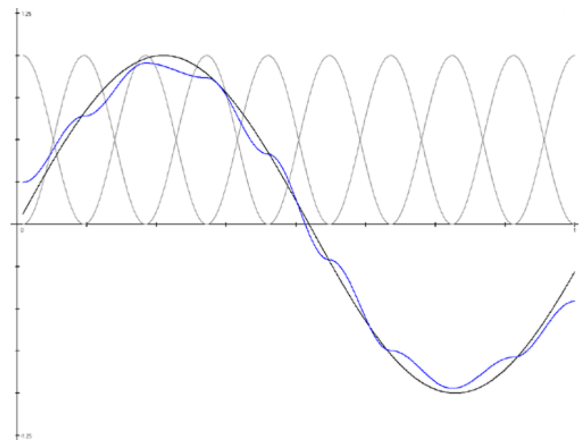
Corollary 1 *[58] Let f be any continuous function on $\mathcal{D}$ and $\{\mathbf{A}_i\}_n$ be a sequence of uniform fuzzy partitions of $\mathcal{D}$, one for each n. Let $\{f^F_n\}_n$ be the sequence of the* *inverse F-transforms, each with respect to a given fuzzy partition. Then for any $\varepsilon > 0$ there exists $n_\varepsilon > 2$ such that for each $n > n_\varepsilon$ and for all $x \in \mathcal{D}$*

$$|f(x) - f_n^F(x)| \leq \varepsilon. \tag{2.2.19}$$

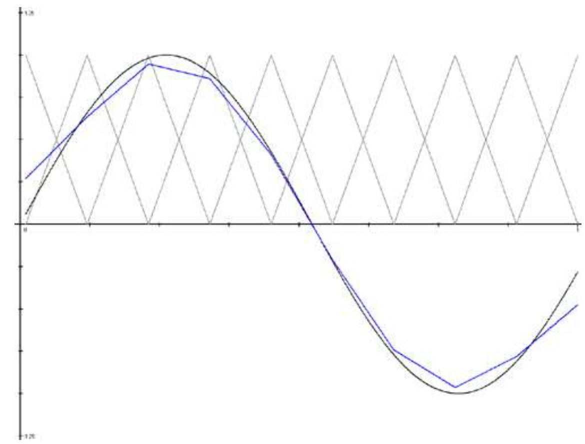
Corollary 2 *[58] Let the assumptions of Corollary 1 be fulfilled. Then the sequence of inverse F-transforms $\{f_n^F\}_n$ uniformly converges to f.*

The assumption of uniformity of fuzzy partitions in Corollary 1 can be avoided. The crucial point is to require such fuzzy partitions that $h_i \to 0$ for $n \to \infty$ and $i = 1, \ldots n - 1$.

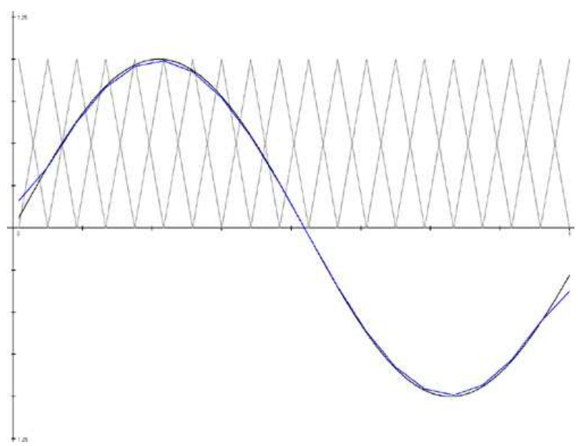
The uniform convergence property is visually demonstrated on Figure 2.2.



(a) Fuzzy partition composed of 10 sinusoidal shaped basic functions



(b) Fuzzy partition composed of 10 triangular shaped basic functions



(c) Fuzzy partition composed of 20 triangular shaped basic functions

Figure 2.2: Function $f(x) = \sin(6x)$ (depicted by black color) on $\mathcal{D} = [0, 1]$ and its inverse F-transform (depicted by blue color) w.r.t. a given uniform fuzzy partition (depicted by grey color).

Chapter 3

F-transform for Functions with More Variables

In this chapter, we focus on an extension of the F-transform technique for functions with more variables. This will be used in up-coming applications.

3.1 F-transform for Functions with Two Variables

First of all, we present an extension for functions with two variables [73], which is the most visual case and allows us to present some figures.

3.1.1 Fuzzy Partition

We will consider a rectangle $\mathcal{D}^2 = [a, b] \times [c, d]$ as a common domain of all real-valued functions in this section. The main idea consists in a construction of two fuzzy partitions, one of $[a, b]$ and one of $[c, d]$.

Definition 24 [73] Let a fuzzy partition of $[a, b]$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \underset{\sim}{\subset} [a, b], n > 2$ and let a fuzzy partition of $[c, d]$ be given by basic functions $\mathbf{B}_1, \ldots, \mathbf{B}_n \underset{\sim}{\subset} [c, d], m > 2$. Then, the *fuzzy partition of* $\mathcal{D}^2$ is given by the fuzzy Cartesian product $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$ with respect to the product t-norm of these two fuzzy partitions. If both fuzzy partitions of particular axes are uniform (with a symmetry) then, the the overall fuzzy partition is also uniform (with a symmetry).

Remark 8 *Throughout the section, we will use the following notation. The nodes of the fuzzy partition of* $[a,b]$ *will be denoted by* $c_0^x = c_1^x < \cdots < c_n^x = c_{n+1}^x$ *and the nodes of the fuzzy partition of* $[c,d]$ *will be denoted by* $c_0^y = c_1^y < \cdots < c_m^y = c_{m+1}^y$. *Furthermore, we denote* $h_i^x = c_{i+1}^x - c_i^x$, $i = 0,\ldots,n$ *and* $h_j^y = c_{j+1}^y - c_j^y$, $j = 0,\ldots,m$ *and analogously* $h^x = (b-a)/(n-1)$ *and* $h^y = (d-c)/(m-1)$ *in the case of a uniform fuzzy partition.*

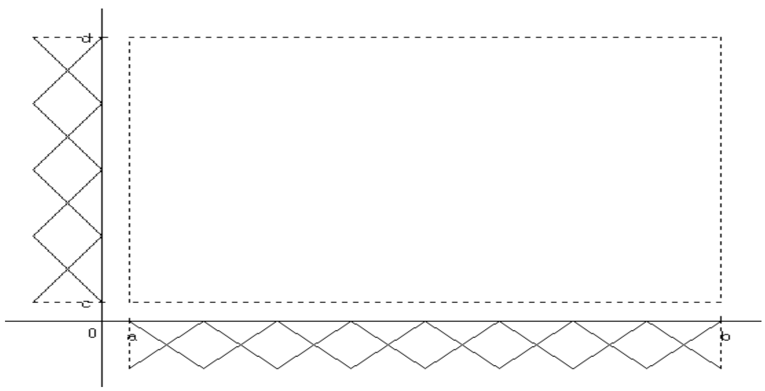


Figure 3.1: Uniform fuzzy partition of $\mathcal{D}^2$ comprised from triangular shaped basic functions on both axes.

Similarly to Lemma 2, we can state the following one confirming the independence of double definite integral of product of two uniform basic functions on their shapes.

Lemma 10 *[73] Let a uniform fuzzy partition of* $\mathcal{D}^2$ *be given by* $\{\mathbf{A}_1,\ldots,\mathbf{A}_n\} \times_\odot \{\mathbf{B}_1,\ldots,\mathbf{B}_m\}$. *Then*

$$\int_c^d \int_a^b \mathbf{A}_1(x)\mathbf{B}_1(y)dxdy = \int_c^d \int_a^b \mathbf{A}_1(x)\mathbf{B}_m(y)dxdy =$$

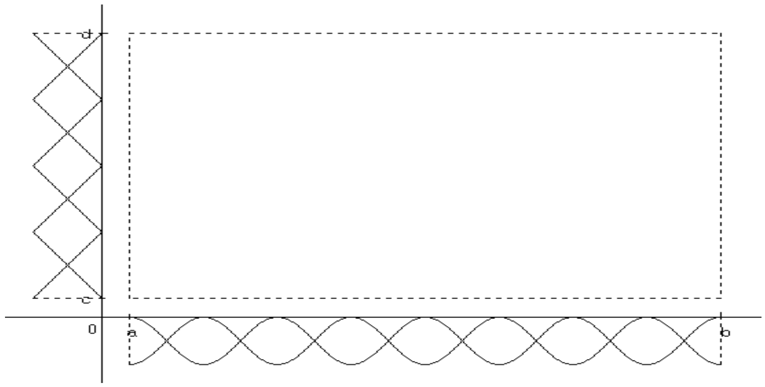


Figure 3.2: Uniform fuzzy partition of $\mathcal{D}^2$ comprised from sinusoidal and triangular shaped basic functions.

$$= \int_c^d \int_a^b \mathbf{A}_n(x)\mathbf{B}_1(y)dxdy = \int_c^d \int_a^b \mathbf{A}_n(x)\mathbf{B}_m(y)dxdy =$$
$$= \frac{h^x h^y}{4}$$

and for $i = 2, \ldots, n-1$ *and* $j = 2, \ldots, m-1$

$$\int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_1(y)dxdy = \int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_m(y)dxdy =$$
$$= \int_c^d \int_a^b \mathbf{A}_1(x)\mathbf{B}_j(y)dxdy = \int_c^d \int_a^b \mathbf{A}_n(x)\mathbf{B}_j(y)dxdy =$$
$$= \frac{h^x h^y}{2}$$

and

$$\int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_j(y)dxdy = h^x h^y.$$

PROOF: By Lemma 2,

$$\int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_j(y)dxdy = \int_c^d \mathbf{B}_j(y) \int_a^b \mathbf{A}_i(x)dxdy = \int_c^d \mathbf{B}_j(y)h^x dy = h^x h^y,$$

for $i = 2, \ldots, n-1$ and $j = 2, \ldots, m-1$.

Analogously we have for $i = 2, \ldots, n-1$ and $j = 2, \ldots, m-1$

$$\int_c^d \int_a^b \mathbf{A}_1(x)\mathbf{B}_j(y)dxdy = \frac{h^x h^y}{2},$$

$$\int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_1(y)dxdy = \frac{h^x h^y}{2}$$

as well as

$$\int_c^d \int_a^b \mathbf{A}_n(x)\mathbf{B}_j(y)dxdy = \frac{h^x h^y}{2},$$

$$\int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_n(y)dxdy = \frac{h^x h^y}{2}.$$

Finally, for $i \in \{1, n\}$ and $j \in \{1, m\}$

$$\int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_j(y)dxdy = \frac{h^x h^y}{4}.$$

$\square$

This lemma will be shorten and generalized to a more-dimensional case in section 3.2.

Due to Lemma 3, we can formulate an analogous lemma even for fuzzy partitions with a symmetry.

Lemma 11 *Let a fuzzy partition with a symmetry of $\mathcal{D}^2$ be given by $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$. Then*

$$\int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_j(y)dxdy = \frac{(h^x_{i-1} + h^x_i)(h^y_{j-1} + h^y_j)}{4}. \tag{3.1.1}$$

PROOF: The proof is analogously to the proof of Lemma 10 and uses Lemma 3. Therefore the proof is omitted. $\square$

3.1.2 Direct and Inverse F-transform

Analogously to the one-dimensional case, we introduce the direct F-transform as a mapping from the space of continuous functions $\mathcal{C}(\mathcal{D}^2)$ to the space of real matrices of $n \times m$ type.

Definition 25 [73] Let a fuzzy partition of $\mathcal{D}^2$ be given by $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$ and let $f \in \mathcal{C}(\mathcal{D}^2)$. We say that a real matrix $F^2[f] = [F_{ij}]_{n\times m}$ given by

$$F_{ij} = \frac{\int_c^d \int_a^b f(x,y)\mathbf{A}_i(x)\mathbf{B}_j(y)dxdy}{\int_c^d \int_a^b \mathbf{A}_i(x)\mathbf{B}_j(y)dxdy}, \; i = 1, \ldots, n, \; j = 1, \ldots, m \tag{3.1.2}$$

is the *F-transform* of f with respect to the given fuzzy partition. The reals F_{ij} are the *components* of the F-transform of f.

By Lemma 10, we can write

$$F_{11} = \frac{4}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_1(x)\mathbf{B}_1(y)dxdy,$$

$$F_{1m} = \frac{4}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_1(x)\mathbf{B}_m(y)dxdy,$$

$$F_{n1} = \frac{4}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_n(x)\mathbf{B}_1(y)dxdy,$$

$$F_{nm} = \frac{4}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_n(x)\mathbf{B}_m(y)dxdy,$$

and for $i = 2, ..., n-1$ and $j = 2, ..., m-1$:

$$F_{i1} = \frac{2}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_i(x)\mathbf{B}_1(y)dxdy,$$

$$F_{im} = \frac{2}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_i(x)\mathbf{B}_m(y)dxdy,$$

$$F_{1j} = \frac{2}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_1(x)\mathbf{B}_j(y)dxdy,$$

$$F_{nj} = \frac{2}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_n(x)\mathbf{B}_j(y)dxdy,$$

$$F_{ij} = \frac{1}{h^x h^y} \int_c^d \int_a^b f(x,y)\mathbf{A}_i(x)\mathbf{B}_j(y)dxdy,$$

In the case of a uniform fuzzy partition of $\mathcal{D}^2$.

By Lemma 11, for $i = 1, ..., n$ and $j = 1, ..., m$ we can write

$$F_{ij} = \frac{4}{(h_{i-1}^x + h_i^x)(h_{j-1}^y + h_j^y)} \int_c^d \int_a^b f(x,y)\mathbf{A}_i(x)\mathbf{B}_j(y)dxdy$$

in the case of a fuzzy partition with a symmetry.

Now, let us present a lemma confirming that the components of the F-transform of $f \in \mathcal{C}(\mathcal{D}^2)$ given by (3.1.2) are (similarly to the one-dimensional case, see Lemma 5 and Lemma 6) equal to precise values of f at the respective nodes of the fuzzy partition up to a certain accuracy.

Lemma 12 *Let a uniform fuzzy partition of $\mathcal{D}^2$ be given by $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$ and let $f \in \mathcal{C}(\mathcal{D}^2)$ be a twice continuously differentiable function on $(a,b) \times (c,d)$. Then for each $i = 1, \ldots, n$ and for each $j = 1, \ldots, m$*

$$F_{ij} = f(c_i^x, c_j^y) + O(\max((h^x)^2, (h^y)^2)). \tag{3.1.3}$$

PROOF: For $i = 2, \ldots, n-1$, and $j = 2, \ldots, m-1$

$$F_{ij} = \frac{1}{h^x h^y} \int_{c_{j-1}^y}^{c_{j+1}^y} \int_{c_{i-1}^x}^{c_{i+1}^x} f(x,y)\mathbf{A}_i(x)\mathbf{B}_j(y)dxdy =$$

$$\frac{1}{h^x h^y} \int_{c_{j-1}^y}^{c_{j+1}^y} \mathbf{B}_j(y) \int_{c_{i-1}^x}^{c_{i+1}^x} f(x,y)\mathbf{A}_i(x)dxdy$$

which is due to the trapezium formula equal to

$$\frac{1}{h^x h^y}\int\limits_{c_{j-1}^y}^{c_{j+1}^y} \mathbf{B}_j(y)$$
$$\left[\frac{h^x}{2}\left(f(c_{i-1}^x,y)\mathbf{A}_i(c_{i-1}^x)+2f(c_i^x,y)\mathbf{A}_i(c_i^x)++f(c_{i+1}^x,y)\mathbf{A}_i(c_{i+1}^x)\right)+O((h^x)^3)\right]dy$$

and because of the properties of the basic functions it is equal to

$$\frac{1}{h^x h^y}\int\limits_{c_{j-1}^y}^{c_{j+1}^y} \mathbf{B}_j(y)(h^x f(c_i^x,y)+O((h^x)^3))dy =$$
$$\frac{1}{h^y}\int\limits_{c_{j-1}^y}^{c_{j+1}^y} \mathbf{B}_j(y)f(c_i^x,y)dy+\frac{1}{h^x h^y}\int\limits_{c_{j-1}^y}^{c_{j+1}^y} \mathbf{B}_j(y)O((h^x)^3)dy.$$

Again, using the trapezium formula we get

$$\frac{1}{h^y}\left[\frac{h^y}{2}(f(c_i^x,c_{j-1}^y)\mathbf{B}_j(c_{j-1}^y)+2f(c_i^x,c_j^y)\mathbf{B}_j(c_j^y)+f(c_i^x,c_{j+1}^y)\mathbf{B}_j(c_{j+1}^y))\right]+$$

$$\frac{O((h^y)^3)}{h^y}+\frac{O((h^x)^3)}{h^x h^y}\int\limits_{c_{j-1}^y}^{c_{j+1}^y} \mathbf{B}_j(y)dy = f(c_i^x,c_j^y)+O((h^x)^2)+\frac{O((h^y)^3)}{h^x h^y}h^y$$

which is finally

$$f(c_i^x,c_j^y)+O(\max((h^x)^2,(h^y)^2)).$$

For the remaining cases, the proof is analogous. $\square$

Lemma 12 can also be generalized for a fuzzy partition with a symmetry.

Lemma 13 *Let a fuzzy partition with a symmetry of* $\mathcal{D}^2$ *be given by* $\{\mathbf{A}_1,\dots,\mathbf{A}_n\}\times_\odot \{\mathbf{B}_1,\dots,\mathbf{B}_m\}$ *and let* $f\in\mathcal{C}(\mathcal{D}^2)$ *be a twice continuously differentiable function on* $(a,b)\times(c,d)$. *Then for each* $i=1,\dots,n$ *and* $j=1,\dots,m$

$$F_{ij}=f(c_i^x,c_j^y)+O(max((h_{i-1}^x)^2,(h_i^x)^2,(h_{j-1}^y)^2,(h_j^y)^2)). \tag{3.1.4}$$

PROOF: By Lemma 11 we can write

$$F_{ij} = \frac{4}{(h^x_{i-1}+h^x_i)(h^y_{j-1}+h^y_j)} \int_c^d \int_a^b f(x,y)\mathbf{A}_i(x)\mathbf{B}_j(y)dxdy$$

which is equal to

$$\frac{4}{(h^x_{i-1}+h^x_i)(h^y_{j-1}+h^y_j)} \int_{c^y_{j-1}}^{c^y_{j+1}} \mathbf{B}_j(y) \left(\int_{c^x_{i-1}}^{c^x_i} f(x,y)\mathbf{A}_i(x)dx + \int_{c^x_i}^{c^x_{i+1}} f(x,y)\mathbf{A}_i(x)dx \right) dy$$

which is (see the proof of Lemma 6) equal to

$$\frac{4}{(h^x_{i-1}+h^x_i)(h^y_{j-1}+h^y_j)} \int_{c^y_{j-1}}^{c^y_{j+1}} \mathbf{B}_j(y) \left[\frac{(h^x_{i-1}+h^x_i)}{2} f(c^x_i, y) + O((h^x_{i-1})^3 + (h^x_i)^3) \right] dy =$$

$$\frac{2}{(h^y_{j-1}+h^y_j)} \int_{c^y_{j-1}}^{c^y_{j+1}} \mathbf{B}_j(y) f(c^x_i, y)dy + \frac{O((h^x_{i-1})^2 + (h^x_i)^2)}{(h^y_{j-1}+h^y_j)} \int_{c^y_{j-1}}^{c^y_{j+1}} \mathbf{B}_j(y)dy$$

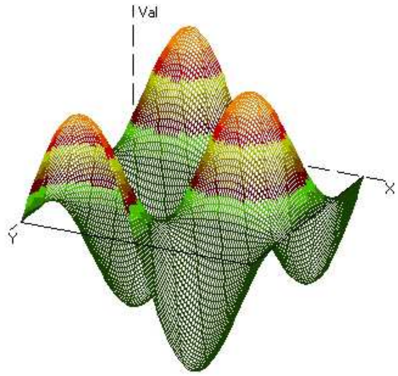
which is by Lemma 6 and by Lemma 3 equal to

$$f(c^x_i, c^y_j) + O(\max((h^x_{i-1})^2, (h^x_i)^2, (h^y_{j-1})^2, (h^y_j)^2)).$$

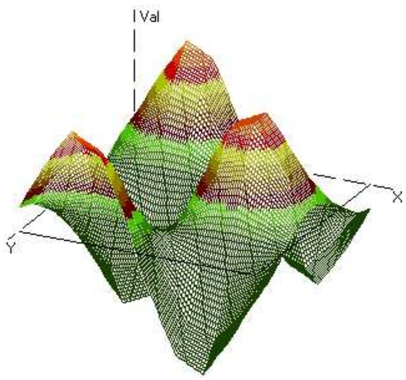
$\square$

Lemma 12 as well as Lemma 13 can be formulated for an arbitrary number of variables and such extensions will be introduced later on. However, the proofs of the extensions using the same technique would not be very transparent due to a high number of indices.

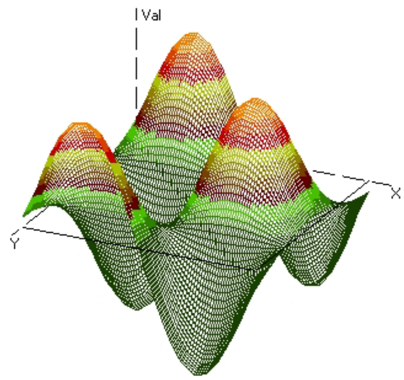
For those situations when the knowledge of f is partial (e.g. given by some measurements) and we know f only at some nodes $(p_k, q_k) \in \mathcal{D}^2$, the direct F-transform is defined as follows.



(a) $f(x, y) = \sin(x) \cdot \cos(y)$



(b) The inverse F-transform w.r.t. a uniform fuzzy partition which is comprised of 10 triangular shaped basic functions on each axis



(c) The inverse F-transform w.r.t. a uniform fuzzy partition which is comprised of 20 triangular shaped basic functions on each axis

Figure 3.3: An illustration of the uniform convergence of the inverse F-transform to the original function $f(x, y) = \sin(x) \cdot \cos(y)$ on $\mathcal{D}^2 = [0, 2\pi]^2$.

Definition 26 Let a fuzzy partition of $\mathcal{D}^2$ be given by $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$ and let a function $f : \mathcal{D}^2 \to \mathbb{R}$ be known at nodes $(p_1, q_1), \ldots, (p_N, q_N)$ such that for each (i, j) where $i = 1, \ldots, n$ and $j = 1, \ldots, m$, there exists $k = 1, \ldots, N$: $\mathbf{A}_i(p_k)\mathbf{B}_j(q_k) > 0$. We say that a real matrix $F^2[f] = [F_{ij}]_{n \times m}$ given by

$$F_{ij} = \frac{\sum_{k=1}^{N} f(p_k, q_k)\mathbf{A}_i(p_k)\mathbf{B}_j(q_k)}{\sum_{k=1}^{N} \mathbf{A}_i(p_k)\mathbf{B}_j(q_k)},\ i = 1, \ldots, n,\ j = 1, \ldots, m \tag{3.1.5}$$

is the *discrete (direct) F-transform* of f with respect to the given fuzzy partition.

Definition 27 Let $F^2[f]$ be the F-transform of a function $f \in \mathcal{C}(\mathcal{D}^2)$ with respect to a given fuzzy partition $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_\odot \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$. Then the function

$$f^F_{n,m}(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{m} F_{ij} \mathbf{A}_i(x) \mathbf{B}_j(y) \tag{3.1.6}$$

is called the *inverse F-transform* of f.

The Inverse F-transform is defined analogously to the one-dimensional case.

A uniform convergence of the inverse F-transform to the original function, as well as some other properties, will be proved in Section 3.2

3.2 Fuzzy Transform for Functions with $r \in \mathbb{N}$ Variables

This section provides the full extension of the original concept of the F-transform method to the space of functions with an arbitrary finite number of arguments. The previous sections devoted to the two-dimensional case was provided because of clear motivation, better visualization, possibility to provide the reader with figures and more transparent proofs for some properties (see Lemmas 12 and 13).

3.2.1 Fuzzy Partition

We will denote the common domain of all functions with r variables by $\mathcal{D}^r = [a_1, b_1] \times \cdots \times [a_r, b_r]$, vectors $\mathbf{x} = (x^1, \ldots, x^r)$ will denote elements of $\mathcal{D}^r$.

Definition 28 [74] Let a fuzzy partition of $[a_j, b_j]$ be given by basic functions $\mathbf{A}^j_1, \ldots, \mathbf{A}^j_{n_j} \subsetapprox [a_j, b_j], n > 2$ for $j = 1, \ldots, r$. Then the *fuzzy partition of* $\mathcal{D}^r$ is given by the fuzzy Cartesian product $\{\mathbf{A}^1_1, \ldots, \mathbf{A}^1_{n_1}\} \times_\odot \{\mathbf{A}^2_1, \ldots, \mathbf{A}^2_{n_2}\} \times_\odot \cdots \times_\odot \{\mathbf{A}^r_1, \ldots, \mathbf{A}^r_{n_r}\}$ with respect to the product t-norm of these r fuzzy partitions. If all fuzzy partitions of particular axes are uniform (with a symmetry) then also the overall fuzzy partition is uniform (with a symmetry).

Remark 9 *Throughout the section, we will use the following notation. The nodes of the fuzzy partition of* $[a_j, b_j]$ *will be denoted by* $c_0^j = c_1^j < \cdots < c_{n_j}^j = c_{n_j+1}^j$. *Furthermore, we denote* $h_{i_j}^j = c_{i_j+1}^j - c_{i_j}^j$, *for* $i_j = 0, \ldots, n_j$ *and* $j = 0, \ldots, r$ *and analogously* $h^j = (b_j - a_j)/(n_j - 1)$ *in the case of a uniform fuzzy partition.*

Now we can present the following lemma generalizing and simplifying Lemma 10.

Lemma 14 *[74] Let* $\{\mathbf{A}_1^1, \ldots, \mathbf{A}_{n_1}^1\} \times_{\odot} \{\mathbf{A}_1^2, \ldots, \mathbf{A}_{n_2}^2\} \times_{\odot} \cdots \times_{\odot} \{\mathbf{A}_1^r, \ldots, \mathbf{A}_{n_r}^r\}$ *be a uniform fuzzy partition of* $\mathcal{D}^r$. *Then*

$$\int_{\mathcal{D}^r} \mathbf{A}_{i_1}^1(x^1) \cdots \mathbf{A}_{i_r}^r(x^r) d\mathbf{x} = \frac{h^1 \cdots h^r}{2^\alpha}, \tag{3.2.1}$$

where α *is the frequency of occurrence* $i_j = 1$ *or* $i_j = n_j$ *for all* $j = 1, \ldots, r$.

PROOF: Due to Lemma 2

$$\int_{a_j}^{b_j} \mathbf{A}_{i_j}^j(x^j) dx^j = h^j \quad \text{for } i_j = 2, \ldots, n_j - 1,$$

and

$$\int_{a_j}^{b_j} \mathbf{A}_1^j(x^j) dx^j = \int_{a_j}^{b_j} \mathbf{A}_{n_j}^j(x^j) dx^j = \frac{h^j}{2}.$$

The remainder of the proof is just a technical exercise (see the proof of Lemma 10) and therefore is omitted. □

Which can be for non-uniform basic functions formalized as follows.

Lemma 15 *Let* $\{\mathbf{A}_1^1, \ldots, \mathbf{A}_{n_1}^1\} \times_{\odot} \{\mathbf{A}_1^2, \ldots, \mathbf{A}_{n_2}^2\} \times_{\odot} \cdots \times_{\odot} \{\mathbf{A}_1^r, \ldots, \mathbf{A}_{n_r}^r\}$ *be a fuzzy partition with a symmetry of* $\mathcal{D}^r$. *Then*

$$\int_{\mathcal{D}^r} \mathbf{A}_{i_1}^1(x^1) \cdots \mathbf{A}_{i_r}^r(x^r) d\mathbf{x} = \frac{\prod_{j=1}^{r} (h_{i_j-1}^j + h_{i_j}^j)}{2^r}. \tag{3.2.2}$$

PROOF: Due to Lemma 3

$$\int_{a_j}^{b_j} \mathbf{A}^j_{i_j}(x^j)dx^j = \frac{(h^j_{i_j-1} + h^j_{i_j})}{2} \quad \text{for } i_j = 1, \ldots, n_j,$$

which proves the lemma. □

3.2.2 Direct F-transform

Definition 29 Let $f \in \mathcal{C}(\mathcal{D}^r)$ and let $\{\mathbf{A}^1_1, \ldots, \mathbf{A}^1_{n_1}\} \times_\odot \{\mathbf{A}^2_1, \ldots, \mathbf{A}^2_{n_2}\} \times_\odot \cdots \times_\odot \{\mathbf{A}^r_1, \ldots, \mathbf{A}^r_{n_r}\}$ be a fuzzy partition of $\mathcal{D}^r$. We say that a ν-tuple $F^{(r)}[f] = [F_{i_1 \cdots i_r}]$ where $\nu = (n_1 \cdot n_2 \cdot \ldots \cdot n_r)$ of real numbers is *the direct F-transform* of f with respect to the given fuzzy partition if

$$F_{i_1 \cdots i_r} = \frac{\int_{a_r}^{b_r} \cdots \int_{a_1}^{b_1} f(x^1, \ldots, x^r)\mathbf{A}^1_{i_1}(x^1) \cdots \mathbf{A}^r_{i_r}(x^r)dx^1 \cdots dx^r}{\int_{a_r}^{b_r} \cdots \int_{a_1}^{b_1} \mathbf{A}^1_{i_1}(x^1) \cdots \mathbf{A}^r_{i_r}(x^r)dx^r \cdots dx^r} \tag{3.2.3}$$

for each r-tuple $i_1 \cdots i_r$.

Remark 10 *Number ν will denote the following product $\nu = (n_1 \cdot n_2 \cdot \ldots \cdot n_r)$ throughout the entire thesis.*

It is easy to see that if the fuzzy partition is fixed, then the direct F-transform as a mapping from $\mathcal{C}(\mathcal{D}^r)$ to $\mathbb{R}^\nu$ is linear.

Lemma 16 *Let $\{\mathbf{A}^1_1, \ldots, \mathbf{A}^1_{n_1}\} \times_\odot \{\mathbf{A}^2_1, \ldots, \mathbf{A}^2_{n_2}\} \times_\odot \cdots \times_\odot \{\mathbf{A}^r_1, \ldots, \mathbf{A}^r_{n_r}\}$ be a fuzzy partition of $\mathcal{D}^r$ and let f, g, h be continuous functions on $\mathcal{D}^r$ such that $h = \alpha f + \beta g$ where α, β are real numbers. Then the following equality holds*

$$F^{(r)}[\alpha f + \beta g] = \alpha F^{(r)}[f] + \beta F^{(r)}[g]. \tag{3.2.4}$$

PROOF: By direct computation one gets the proof using the fact that the definite integral is a linear mapping, see the proof of Lemma 4. □

Definition 30 Let $\{\mathbf{A}_1^1, \ldots, \mathbf{A}_{n_1}^1\} \times_\odot \{\mathbf{A}_1^2, \ldots, \mathbf{A}_{n_2}^2\} \times_\odot \cdots \times_\odot \{\mathbf{A}_1^r, \ldots, \mathbf{A}_{n_r}^r\}$ be a fuzzy partition of $\mathcal{D}^r$ and let a function $f : \mathcal{D}^r \to \mathbb{R}$ be known at nodes $(p_1^1, \ldots, p_1^r), \ldots, (p_N^1 \ldots, p_N^r)$ such that for each $(i_1, \cdots, i_r)$ where $i_j = 1, \ldots, n_j$ and $j = 1, \ldots, r$, there exists $k = 1, \ldots, N$: $\mathbf{A}_{i_1}^1(p_k^1) \cdots \mathbf{A}_{i_r}^r(p_k^r) > 0$. We say that the ν-tuple $[F_{i_1 \cdots i_r}]$ of real numbers is *the discrete direct F-transform* of f with respect to the given fuzzy partition if

$$F_{i_1 \cdots i_r} = \frac{\sum_{k=1}^N f(p_k^1, \ldots, p_k^r)\mathbf{A}_{i_1}^1(p_k^1) \cdots \mathbf{A}_{i_r}^r(p_{i_r}^r)}{\sum_{k=1}^N \mathbf{A}_{i_1}^1(p_k^1) \cdots \mathbf{A}_{i_r}^r(p_{i_r}^r)} \tag{3.2.5}$$

for each r-tuple $i_1 \cdots i_r$.

Lemma 17 *Let $f \in \mathcal{C}(\mathcal{D}^r)$ be a twice continuously differentiable function on $(a_1, b_1) \times \cdots \times (a_r, b_r)$ and $\{\mathbf{A}_1^1, \ldots, \mathbf{A}_{n_1}^1\} \times_\odot \{\mathbf{A}_1^2, \ldots, \mathbf{A}_{n_2}^2\} \times_\odot \cdots \times_\odot \{\mathbf{A}_1^r, \ldots, \mathbf{A}_{n_r}^r\}$ be a uniform fuzzy partition of $\mathcal{D}^r$. Then*

$$F_{i_1 \ldots i_r} = f(c_{i_1}^1, \ldots, c_{i_r}^r) + O\left(\bigvee_{j=1}^r (h^j)^2\right) \tag{3.2.6}$$

for each $i_j = 1, \ldots, n_j$ where $j = 1, \ldots, r$.

PROOF: The proof uses the same technique as the proof of Lemma 12 and therefore it is omitted. □

Finally, we can extend Lemma 13 as follows.

Lemma 18 *Let $f \in \mathcal{C}(\mathcal{D}^r)$ be a twice continuously differentiable function on $(a_1, b_1) \times \cdots \times (a_r, b_r)$ and $\{\mathbf{A}_1^1, \ldots, \mathbf{A}_{n_1}^1\} \times_\odot \{\mathbf{A}_1^2, \ldots, \mathbf{A}_{n_2}^2\} \times_\odot \cdots \times_\odot \{\mathbf{A}_1^r, \ldots, \mathbf{A}_{n_r}^r\}$ be a fuzzy partition with a symmetry of $\mathcal{D}^r$. Then*

$$F_{i_1 \ldots i_r} = f(c_{i_1}^1, \ldots, c_{i_r}^r) + O\left(\max\left(\bigvee_{j=1}^r (h_{i_j-1}^j)^2, \bigvee_{j=1}^r (h_{i_j}^j)^2\right)\right) \tag{3.2.7}$$

for each $i_j = 1, \ldots, n_j$ where $j = 1, \ldots, r$.

PROOF: The proof uses the same technique as the proof of Lemma 13 and therefore it is omitted. □

Since the notation of the equality of the components of the F-transform to the precise function values at nodes of a given fuzzy partition up to a certain accuracy is becoming more complicated and less transparent when the number of variables increase, we can state the following simplification.

Corollary 3 *Let all the assumptions of Lemma 18 be fulfilled and let us denote*

$$h_{max} = \max\{h^j_{i_j} | i_j \in \{1, \ldots, n_j - 1\};\ j \in \{1, \ldots, r\}\}.$$

Then

$$F_{i_1 \ldots i_r} = f(c^1_{i_1}, \ldots, c^r_{i_r}) + O((h_{max})^2) \tag{3.2.8}$$

for each $i_j = 1, \ldots, n_j$ *where* $j = 1, \ldots, r$.

Since any uniform fuzzy partition is only a special case of the fuzzy partition with a symmetry, the previous corollary simplifies the method of expressing the accuracy of the F-transform for both - systems of non-uniform and even uniform basic functions.

3.2.3 Inverse F-transform

Definition 31 Let $F^{(r)}[f]$ be the direct F-transform of $f \in \mathcal{C}(\mathcal{D}^r)$ with respect to a given fuzzy partition $\{\mathbf{A}^1_1, \ldots, \mathbf{A}^1_{n_1}\} \times_\odot \{\mathbf{A}^2_1, \ldots, \mathbf{A}^2_{n_2}\} \times_\odot \cdots \times_\odot \{\mathbf{A}^r_1, \ldots, \mathbf{A}^r_{n_r}\}$. Then function

$$f^F_{n_1,\ldots,n_r}(x^1, \ldots, x^r) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_r=1}^{n_r} F_{i_1 \cdots i_r} \mathbf{A}^1_{i_1}(x^1) \cdots \mathbf{A}^r_{i_r}(x^r) \tag{3.2.9}$$

is called the *inverse F-transform* of f.

Similarly to Lemma 9 we can formulate the following one claiming the one-to-one correspondence of the set $\mathbb{R}^\nu$ and the space $lin_\nu\{\mathbf{A}^j_{i_j}\}$ of linear combinations of ν fuzzy sets forming a fuzzy partition of $\mathcal{D}^r$.

Lemma 19 *Let* $\{\mathbf{A}_1^1, \ldots, \mathbf{A}_{n_1}^1\} \times_{\odot} \{\mathbf{A}_1^2, \ldots, \mathbf{A}_{n_2}^2\} \times_{\odot} \cdots \times_{\odot} \{\mathbf{A}_1^r, \ldots, \mathbf{A}_{n_r}^r\}$ *be a fuzzy partition of* $\mathcal{D}^r$*. Then each* $q \in lin_\nu\{\mathbf{A}_{i_j}^j\}_{n_j}$ *i.e.*

$$q(x^1, \ldots, x^r) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_r=1}^{n_r} Q_{i_1 \cdots i_r} \mathbf{A}_{i_1}^1(x^1) \cdots \mathbf{A}_{i_r}^r(x^r)$$

is uniquely determined by the ν*-tuple* $[Q_{i_1 \ldots i_r}]$ *of real numbers.*

for $Q_1, \ldots, Q_n, R_1, \ldots, R_n \in \mathbb{R}$ such that there exists i: $Q_i \neq R_i$. For that index i, $\mathbf{A}_i(c_i) = 1$, whence $\mathbf{A}_j(c_i) = 0$ for $j \neq i$. This implies $Q_i = R_i$ which contradicts the assumption.

PROOF: By contradiction. Suppose that

$$\sum_{i_1=1}^{n_1} \cdots \sum_{i_r=1}^{n_r} Q_{i_1 \cdots i_r} \mathbf{A}_{i_1}^1(x^1) \cdots \mathbf{A}_{i_r}^r(x^r) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_r=1}^{n_r} R_{i_1 \cdots i_r} \mathbf{A}_{i_1}^1(x^1) \cdots \mathbf{A}_{i_r}^r(x^r)$$

for $Q_{i_1, \ldots, i_r}, R_{i_1, \ldots, i_r} \in \mathbb{R}$ where $i_j = 1, \ldots, n_j$ and $j = 1, \ldots, r$ such that there exists an r-tuple $(i_1, \ldots, i_r)$: $Q_{i_1, \ldots, i_r} \neq R_{i_1, \ldots, i_r}$. For that r-tuple,

$$\mathbf{A}_{i_1}^1(c_{i_1}^1) \cdot \ldots \cdot \mathbf{A}_{i_r}^r(c_{i_r}^r) = 1$$

whence

$$\mathbf{A}_{l_1}^1(c_{l_1}^1) \cdot \ldots \cdot \mathbf{A}_{l_r}^r(c_{l_r}^r) = 0$$

for $(i_1, \ldots, i_r) \neq (l_1, \ldots, l_r)$. This implies $Q_{i_1, \ldots, i_r} = R_{i_1, \ldots, i_r}$ which contradicts the assumption. $\square$

As well as in the one-dimensional case, we can state the crucial convergence property as follows.

Theorem 3 *Let* $f \in \mathcal{C}(\mathcal{D}^r)$*. Then for any* $\varepsilon > 0$ *there exist* $n_j(\varepsilon)$ *for* $j = 1, \ldots, r$ *and a fuzzy partition* $\{\mathbf{A}_1^1, \ldots, \mathbf{A}_{n_1(\varepsilon)}^1\} \times_{\odot} \{\mathbf{A}_1^2, \ldots, \mathbf{A}_{n_2(\varepsilon)}^2\} \times_{\odot} \cdots \times_{\odot} \{\mathbf{A}_1^r, \ldots, \mathbf{A}_{n_r(\varepsilon)}^r\}$ *of* $\mathcal{D}^r$ *such that for all* $\mathbf{x} \in \mathcal{D}^r$

$$\mid f(\mathbf{x}) - f_{n_1(\varepsilon), \ldots, n_r(\varepsilon)}^F(\mathbf{x}) \mid < \varepsilon.$$

PROOF: Since f is continuous, for each $\varepsilon > 0$ we can find some $\delta > 0$ such that for all $\mathbf{u}, \mathbf{v} \in \mathcal{D}^r : \| \mathbf{u} - \mathbf{v} \|_\epsilon < \delta$ implies $|f(\mathbf{u}) - f(\mathbf{v})| < \varepsilon$ where $\| \cdot \|_\epsilon$ denotes the Euclidean norm.

Let $\{\mathbf{A}_1^1, \ldots, \mathbf{A}_{n_1(\varepsilon)}^1\} \times_\odot \{\mathbf{A}_1^2, \ldots, \mathbf{A}_{n_2(\varepsilon)}^2\} \times_\odot \cdots \times_\odot \{\mathbf{A}_1^r, \ldots, \mathbf{A}_{n_r(\varepsilon)}^r\}$ be a uniform fuzzy partition of $\mathcal{D}^r$ such that $h^j = (b_j - a_j)/(n_j(\varepsilon) - 1) \leq \delta/\sqrt{r}$ for all $j = 1, \ldots, r$.

Let $\mathbf{t} = (t^1, \ldots, t^r)$ be an arbitrary element of $[c_{i_1}^1, c_{i_1+1}^1] \times \cdots \times [c_{i_r}^r, c_{i_r+1}^r]$ then

$$| f(\mathbf{t}) - F_{i_1 \ldots i_r} | = | f(\mathbf{t}) - \frac{2^\alpha}{h^1 \cdots h^r} \int\limits_{\mathcal{D}^r} f(\mathbf{x}) \mathbf{A}_{i_1}^1(x^1) \cdots \mathbf{A}_{i_r}^r(x^r) d\mathbf{x} | \leq$$

$$\frac{2^\alpha}{h^1 \cdots h^r} \int\limits_{c_{i_r}^r}^{c_{i_r+1}^r} \cdots \int\limits_{c_{i_1}^1}^{c_{i_1+1}^1} | f(\mathbf{t}) - f(\mathbf{x}) | \mathbf{A}_{i_1}^1(x^1) \cdots \mathbf{A}_{i_r}^r(x^r) d\mathbf{x}.$$

Due to the continuity of f we have

$$\frac{2^\alpha}{h_1 \cdots h_r} \cdot \varepsilon \cdot \int\limits_{c_{i_r}^r}^{c_{i_r+1}^r} \cdots \int\limits_{c_{i_1}^1}^{c_{i_1+1}^1} | f(\mathbf{t}) - f(\mathbf{x}) | \mathbf{A}_{i_1}^1(x^1) \cdots \mathbf{A}_{i_r}^r(x^r) d\mathbf{x}.$$

Due to Lemma 14 the last expressions is equal to

$$\frac{2^\alpha}{h_1 \cdots h_r} \cdot \varepsilon \cdot \frac{h_1 \cdots h_r}{2^\alpha} = \varepsilon.$$

Therefore,

$$| f(\mathbf{t}) - F_{i_1 \ldots i_r} | < \varepsilon.$$

Analogously,

$$| f(\mathbf{t}) - F_{(i_1+1)\dots i_r} |< \varepsilon,$$

$$\dots$$

$$| f(\mathbf{t}) - F_{i_1\dots(i_r+1)} |< \varepsilon,$$

$$\dots$$

$$| f(\mathbf{t}) - F_{(i_1+1)\dots(i_r+1)} |< \varepsilon,$$

and therefore

$$| f(\mathbf{t}) - f^F_{n_1(\varepsilon),\dots,n_r(\varepsilon)}(\mathbf{t}) |=| f(\mathbf{t}) - \sum_{i_1=1}^{n_1(\varepsilon)} \cdots \sum_{i_r=1}^{n_r(\varepsilon)} F_{i_1\dots i_r} \mathbf{A}^1_{i_1}(t^1)\cdots \mathbf{A}^r_{i_r}(t^r) |\leq$$

$$\sum_{i_1=1}^{n_1(\varepsilon)} \cdots \sum_{i_r=1}^{n_r(\varepsilon)} \mathbf{A}^1_{i_1}(t^1)\cdots \mathbf{A}^r_{i_r}(t^r) \mid f(\mathbf{t}) - F_{i_1\dots i_r} |= \varepsilon.$$

Since the argument $\mathbf{t}$ has been chosen arbitrarily, Theorem 3 has been proved. □

Corollary 4 *Let* $f \in \mathcal{C}(\mathcal{D}^r)$ *and let* $\{\{\mathbf{A}^1_1, \dots, \mathbf{A}^1_{n_1}\} \times_\odot \{\mathbf{A}^2_1, \dots, \mathbf{A}^2_{n_2}\} \times_\odot \cdots \times_\odot \{\mathbf{A}^r_1, \dots, \mathbf{A}^r_{n_r}\}\}_{(n_1,n_2,\dots,n_r)}$ *be a sequence of uniform fuzzy partitions of* $\mathcal{D}^r$. *Let* $\{f^F_{n_1,\dots,n_r}\}_{(n_1,n_2,\dots,n_r)}$ *be the sequence of the inverse F-transforms, one for each fuzzy partition. Then for any* $\varepsilon > 0$ *there exist* $n_j(\varepsilon)$ *where* $j = 1, \dots, r$ *such that for each* $n_j > n_j(\varepsilon)$ *and for all* $\mathbf{x} \in \mathcal{D}$

$$| f(\mathbf{x}) - f^F_{n_1,\dots,n_r}(\mathbf{x}) |< \varepsilon.$$

Corollary 5 *Let all the assumptions of Corollary 4 be fulfilled. Then the sequence* $\{f^F_{n_1,\dots,n_r}\}_{(n_1,n_2,\dots,n_r)}$ *of the inverse F-transforms, one for each fuzzy partition, uniformly converges to* f.

Chapter 4

Applications to Partial Differential Equations

4.1 Motivation

Differential equations are used for modelling various physical phenomena. Actually, many problems are dynamic and too complicated in order to develop an accurate differential equation model for such problems requires complex and time consuming algorithms hardly implementable in practice. Thus, a usage of fuzzy mathematics seems to be appropriate for such cases.

Much work has been done in the field of fuzzy differential equations see e.g. [33], [34] or [84]. However, a fuzzy approach to numerical solutions of differential equations has not yet been deeply investigated although some results have been published e.g. in [64].

In [54] I. Perfilieva introduced an application to the Cauchy problem. It has been shown that if the original function is replaced by an approximation model, then a certain simplification of complex computations could be achieved. The main idea consists in applying the fuzzy transform to both sides of an ordinary differential equation of the first order. This transforms the given differential equation to an algebraic one which is solvable by existing methods. Then the obtained numerical solution is transformed back to the space of continuous functions by the inverse F-transform.

4.2 Partial Differential Equations

As known from physics, many problems cannot be expressed by ordinary differential equations thus partial differential are required. Therefore, the extension of the F-transform from the previous Chapter was a necessary step in the approach introduced in [54] to partial differential equations. Results in the field of the application of the F-transform to numerical solutions of partial differential equation have been already published in [72, 74, 75].

4.2.1 Equations of Mathematical Physics

We consider three main types of partial differential equations (the equations of mathematical physics) - heat equation, wave equation and Poisson's equation. Without any loss of generality we consider the 2-dimensional cases of these equations.

Each mentioned partial differential equation is considered on a domain $\mathcal{D}^2 = X \times Y$ where X, Y are closed real intervals. It can be written in the following general shape:

$$\mathbf{L}\left(\frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) = q(x, y) \tag{4.2.1}$$

where $\mathbf{L}$ is a linear form [25, 65]. Moreover, in order to have a unique solution, the specific initial and boundary conditions are given. The open part of the domain will be denoted by $\mathcal{D}_0^2$ which means that $\mathcal{D}_0^2 = \mathcal{D}^2 \setminus \partial\mathcal{D}^2$.

Remark 11 *The heat equation and the wave equation are called evolutionary equations because the second variable is supposed to be a time variable. That is why in* *the next two sections we will consider a domain $\mathcal{D}^2 = X \times T$ and all the symbols y will be replaced by t.*

Our approach consists in applying F-transform (w.r.t. to some given fixed fuzzy partition) to both sides of equation (4.2.1). This leads to an algebraic equation that should be solved. The solution of this algebraic equation gives a discrete representation of an analytical solution of equation (4.2.1) which can be brought back to the space of continuous functions $C(\mathcal{D}^2)$ by the inverse F-transform.

Before we start to analyze concrete types of equations, let us fix a uniform fuzzy partition $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$ of $\mathcal{D}^2$, with an equidistant step h^x on X and with an equidistant step $h^y(h^t)$ on $Y(T)$. All the F-transforms considered in this chapter are computed w.r.t. this fuzzy partition.

4.3 Heat Equation

The first partial differential equation to be studied is the heat equation. An application of F-transform to this equation is investigated in [72]. Let the domain $\mathcal{D}^2$ be a Cartesian product of two real intervals $X = [0, 1]$ and $T = [0, R]$ and let $\mathcal{D}_0'^2 = (0, 1) \times (0, R]$. Let $u : X \times T \to \mathbb{R}$ be a continuous solution to the parabolic equation

$$\frac{\partial u(x,t)}{\partial t} - \alpha \frac{\partial^2 u(x,t)}{\partial x^2} = q(x,t), \quad \alpha \in \mathbb{R}^+, \ (x,t) \in \mathcal{D}_0'^2 \tag{4.3.1}$$

with the following initial and boundary conditions

$$u(x, 0) = f(x), \quad x \in X, \tag{4.3.2}$$

$$u(0,t) = T_1(t), \quad u(1,t) = T_2(t), \quad t \in T. \tag{4.3.3}$$

After applying F-transform, equation (4.3.1) is transferred to the following algebraic equation

$$F^2[u^t] - \alpha F^2[u^{xx}] = F^2[q] \tag{4.3.4}$$

where

$$F^2[u^t] = \begin{bmatrix} U_{11}^t & U_{12}^t & \dots & U_{1m}^t \\ U_{21}^t & U_{22}^t & \dots & U_{2m}^t \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots \\ U_{n1}^t & \dots & \dots & U_{nm}^t \end{bmatrix} \tag{4.3.5}$$

is the matrix of the F-transform components of $\frac{\partial u}{\partial t}$,

$$F^2[u^{xx}] = \begin{bmatrix} U_{11}^{xx} & U_{12}^{xx} & \dots & U_{1m}^{xx} \\ U_{21}^{xx} & U_{22}^{xx} & \dots & U_{2m}^{xx} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots \\ U_{n1}^{xx} & \dots & \dots & U_{nm}^{xx} \end{bmatrix} \tag{4.3.6}$$

is the matrix of the F-transform components of $\frac{\partial^2 u}{\partial x^2}$, and finally

$$F^2[q] = \begin{bmatrix} Q_{11} & Q_{12} & \dots & Q_{1m} \\ Q_{21} & Q_{22} & \dots & Q_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots \\ Q_{n1} & \dots & \dots & Q_{nm} \end{bmatrix} \tag{4.3.7}$$

is the matrix of the F-transform components of q.

Indeed, we are not able to find matrices $F^2[u^t]$ and $F^2[u^{xx}]$ since the partial derivatives $\frac{\partial u}{\partial t}$ and $\frac{\partial^2 u}{\partial x^2}$ are unknown. Therefore, the partial derivatives must be replaced by their finite differences:

$$\frac{\partial u(x,t)}{\partial t} \quad \text{is replaced by} \quad \frac{(u(x,t+h^t) - u(x,t))}{h^t}$$

and analogously

$$\frac{\partial^2 u(x,t)}{\partial x^2} \quad \text{is replaced by} \quad \frac{(u(x+h^x,t) - 2u(x,t) + u(x-h^x,t))}{(h^x)^2}.$$

Next, we can approximate U^t_{ij} as follows:

$$
\begin{aligned}
U^t_{ij} \;=\; & \frac{\int_0^R \int_0^1 \frac{\partial u(x,t)}{\partial t} \mathbf{A}_i(x)\mathbf{B}_j(t)dxdt}{\int_0^R \int_0^1 \mathbf{A}_i(x)\mathbf{B}_j(t)dxdt} \\
& \approx \frac{\int_0^R \int_0^1 \frac{(u(x,t+h^t)-u(x,t))}{h^t} \mathbf{A}_i(x)\mathbf{B}_j(t)dxdt}{\int_0^R \int_0^1 \mathbf{A}_i(x)\mathbf{B}_j(t)dxdt} \\
& = \frac{1}{h^t} \frac{\int_0^R \int_0^1 u(x,t+h^t)\mathbf{A}_i(x)\mathbf{B}_j(t)dxdt}{\int_0^R \int_0^1 \mathbf{A}_i(x)\mathbf{B}_j(t)dxdt} \\
& - \frac{1}{h^t} \frac{\int_0^R \int_0^1 u(x,t)\mathbf{A}_i(x)\mathbf{B}_j(t)dxdt}{\int_0^R \int_0^1 \mathbf{A}_i(x)\mathbf{B}_j(t)dxdt} \\
& = \frac{1}{h_t}(U_{i(j+1)} - U_{ij}). \qquad (4.3.8)
\end{aligned}
$$

Similarly, we obtain

$$U^{xx}_{ij} \approx \frac{1}{(h^x)^2}(U_{(i-1)j} - 2U_{ij} + U_{(i+1)j}). \qquad (4.3.9)$$

By (4.3.8) and (4.3.9), we come to the following recursive equation

$$U_{i(j+1)} = rU_{(i-1)j} + (1-2r)U_{ij} + rU_{(i+1)j} + h^t Q_{ij} \qquad (4.3.10)$$

where $r = (\alpha h^t)/(h^x)^2$ and $i = 2, \ldots, n-1$, $j = 1, \ldots, m-1$.

Remark 12 *The inequality $0 < r \leq 1/2$ must be fulfilled. This condition is associated with the convergence of the numerical solution to the analytical one (see Section 4.6) as well as with stable decay of the errors in the arithmetical operations needed to solve the numerical problem. For more details we refer to [65].*

We apply initial condition (4.3.2) as follows

$$U_{i1} = f((i-1)h^x), \quad i = 1, \ldots, n, \qquad (4.3.11)$$

and boundary conditions (4.3.3) as follows

$$U_{1j} = T_1((j-1)h^t),\; U_{nj} = T_2((j-1)h^t), \quad j = 1, \ldots, m. \qquad (4.3.12)$$

Discrete numerical solution of equation (4.3.1) with initial condition (4.3.2) and boundary conditions (4.3.3) will be given by the following real matrix

$$F^2[u] = \begin{bmatrix} U_{11} & U_{12} & \dots & U_{1m} \\ U_{21} & U_{22} & \dots & U_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots \\ U_{n1} & \dots & \dots & U_{nm} \end{bmatrix} \tag{4.3.13}$$

where each component of the matrix is given either by one of formulas 4.3.11-4.3.12 or it is computed according to recursive formula 4.3.10. This solution is then transferred back to the space $C(\mathcal{D}^2)$ by the inverse F-transform.

4.4 Wave Equation

The second equation of mathematical physics - the wave equation - is considered in this section. An application of the F-transform to this equation has been published in [74, 75] (in [74] even for a more dimensional case). Let the domain $\mathcal{D}^2$ be given by a Cartesian product of two real intervals $X = [0, 1]$ and $T = [0, R]$ and let $\mathcal{D}_0'^2 = (0, 1) \times (0, R]$. Let $u : X \times T \to \mathbb{R}$ be a continuous solution of the following hyperbolic equation

$$\frac{\partial^2 u(x,t)}{\partial x^2} - \alpha^2 \frac{\partial^2 u(x,t)}{\partial t^2} = q(x,t), \quad \alpha \in \mathbb{R},\ (x,t) \in \mathcal{D}_0'^2 \tag{4.4.1}$$

with the following initial and boundary conditions

$$u(x,0) = f(x), \quad \frac{\partial u(x,0)}{\partial t} = g(x), \quad x \in X, \tag{4.4.2}$$

$$u(0,t) = T_1(t), \quad u(1,t) = T_2(t), \quad t \in T. \tag{4.4.3}$$

Similarly to the computation of the heat equation, by applying F-transform we transfer equation (4.4.1) into the following algebraic equation

$$F^2[u^{xx}] - \alpha^2 F^2[u^{tt}] = F^2[q] \tag{4.4.4}$$

where $F^2[u^{xx}]$ and $F^2[q]$ are matrices (4.3.6) and (4.3.7) from the previous subsection, respectively, and where

$$F^2[u^{tt}] = \begin{bmatrix} U_{11}^{tt} & U_{12}^{tt} & \dots & U_{1m}^{tt} \\ U_{21}^{tt} & U_{22}^{tt} & \dots & U_{2m}^{tt} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots \\ U_{n1}^{tt} & \dots & \dots & U_{nm}^{tt} \end{bmatrix}$$

is the matrix of the F-transform components of $\frac{\partial^2 u}{\partial t^2}$.

Analogously to the case of heat equation, values U_{ij}^{xx} are replaced according to (4.3.9) and values U_{ij}^{tt}

$$U_{ij}^{tt} \approx \frac{1}{(h^t)^2}(U_{i(j-1)} - 2U_{ij} + U_{i(j+1)}) \tag{4.4.5}$$

for $i = 1, \dots, n$ and $j = 1, \dots, m$.

By (4.3.9) and (4.4.5) we come to the following recursive equation:

$$\begin{aligned} U_{i(j+1)} = r^2 U_{(i-1)j} + 2(1 - r^2)U_{ij} + \\ + r^2 U_{(i+1)j} - U_{i(j-1)} - (h^t)^2 Q_{ij} \end{aligned} \tag{4.4.6}$$

where $r = h^t/(\alpha h^x)$ and $i = 1, \dots n$, $j = 1, \dots, m$.

Remark 13 *The inequality $0 < r \leq 1$ must be fulfilled. This condition is associated with the convergence of the numerical solution to the analytical one (see Section 4.6) as well as with stable decay of the errors in the arithmetical operations needed to solve the numerical problem. For more details we refer to [65].*

Initial conditions (4.4.2) and boundary conditions (4.4.3) are taken into account as well. The boundary conditions are applied as follows

$$U_{1j} = T_1((j-1)h^t),\ U_{nj} = T_2((j-1)h^t), \quad j = 1, \dots, m. \tag{4.4.7}$$

The first of initial conditions (4.4.2) is applied as follows

$$U_{i1} = f((i-1)h^x). \tag{4.4.8}$$

Expressions U_{i0} for $i = 2, \ldots, n-1$ are last unknown occurring in recursive equation (4.4.6). To determine them, we apply the following finite difference

$$\frac{\partial u}{\partial t} \approx \frac{(u(x, t+h^t) - u(x, t-h^t))}{2h^t} \tag{4.4.9}$$

to the second of initial conditions (4.4.2), which leads to

$$g((i-1)h^x) = \frac{U_{i2} - U_{i0}}{2h^t}$$

which can be rewritten as follows

$$U_{i0} = U_{i2} - 2h^t g((i-1)h^x). \tag{4.4.10}$$

The following matrix

$$F^2[u] = \begin{bmatrix} U_{11} & U_{12} & \ldots & U_{1m} \\ U_{21} & U_{22} & \ldots & U_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ldots & \ldots & \vdots \\ U_{n1} & \ldots & \ldots & U_{nm} \end{bmatrix} \tag{4.4.11}$$

where the components U_{ij} are given either by formulas 4.4.7)-(4.4.10) or they are computed according to (4.4.6), serves as a discrete solution to equation (4.4.1) with initial (4.4.2) and boundary (4.4.3) conditions. This solution is then transferred back to the space $C(\mathcal{D}^2)$ by the inverse F-transform.

4.5 Poisson's Equation

In the case of the Poisson's equation, the finite difference scheme leads to a large set of linear algebraic equations with respect to a complete set of unknowns. There is no step-by-step algorithm analogous to parabolic or hyperbolic equations which computes the unknown parameters, see [25].

Let the domain $\mathcal{D}^2$ be unit cube i.e. a Cartesian product of two real unit intervals $\mathcal{D}$=$[0,1] \times [0,1]$. Let $u(x,y)$ be a continuous solution of the following hyperbolic equation

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = -q(x,y), \quad \alpha \in \mathbb{R}^+,\ (x,y) \in \mathcal{D}_0^2 \tag{4.5.1}$$

with the following Dirichlet boundary condition

$$u(x,y) = g(x,y), \quad (x,y) \in \partial\mathcal{D}^2 \tag{4.5.2}$$

where $\partial\mathcal{D}^2 = \mathcal{D}^2 \setminus \mathcal{D}_0^2$ is a boundary of the domain $\mathcal{D}^2$.

Because both derivatives on the left-hand side of equation (4.5.1) are of the same order we set up the same step on both axes [25] i.e. $h = h^x = h^y$ which yields $m = n$.

Analogously to both previous partial differential equations, we approximate the second order derivatives on the left-hand side by finite differences. Next, we apply the F-transform to both sides of equation (4.5.1) whereby we obtain the following system

$$F^2[u^{xx}] + F^2[u^{yy}] = -F^2[q] \tag{4.5.3}$$

where $F^2[u^{xx}]$ and $F^2[q]$ are matrices (4.3.6) and (4.3.7) from Section 4.4 and where

$$F^2[u^{yy}] = \begin{bmatrix} U_{11}^{yy} & U_{12}^{yy} & \dots & U_{1m}^{yy} \\ U_{21}^{yy} & U_{22}^{yy} & \dots & U_{2m}^{yy} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots \\ U_{n1}^{yy} & \dots & \dots & U_{nm}^{yy} \end{bmatrix}$$

is a matrix of the F-transform components of $\frac{\partial^2 u}{\partial y^2}$.

Again, values U_{ij}^{xx} and values U_{ij}^{yy} are replaced by finite difference as follows

$$U_{ij}^{xx} \approx \frac{1}{h^2}(U_{i(j-1)} - 2U_{ij} + U_{i(j+1)}) \tag{4.5.4}$$

and

$$U_{ij}^{yy} \approx \frac{1}{h^2}(U_{i(j-1)} - 2U_{ij} + U_{i(j+1)}) \tag{4.5.5}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

By (4.5.4) and (4.5.5) we come to the following recursive equation

$$\frac{1}{h^2}(4U_{ij} - U_{(i-1)j} - U_{(i+1)j} - U_{i(j-1)} - U_{i(j+1)}) = Q_{ij} \tag{4.5.6}$$

for $i, j = 2, \ldots, n-1$.

Boundary condition (4.5.2) is applied as follows

$$U_{ij} = G_{ij} \equiv g((i-1)h, (j-1)h), \quad i, j \in \{1, n\}. \tag{4.5.7}$$

Values U_{ij} for $i, j \notin \{1, n\}$ are determined by solving the following system of linear equations

$$K_h u_h = q_h \tag{4.5.8}$$

where u_h and q_h are the following real vectors

$$u_h = [U_{22}, \ldots, U_{2(n-1)}; U_{32}, \ldots, U_{3(n-1)}; \ldots; U_{(n-1)2}, \ldots, U_{(n-1)(n-1)}]^T$$

and

$$q_h = [\mathcal{Q}_{22}, \ldots, \mathcal{Q}_{2(n-1)}; \mathcal{Q}_{32}, \ldots, \mathcal{Q}_{3(n-1)}; \ldots; \mathcal{Q}_{(n-1)2}, \ldots, \mathcal{Q}_{(n-1)(n-1)}]^T,$$

respectively and where

$$
\begin{aligned}
\mathcal{Q}_{ij} &= h^2 Q_{ij}, \\
\mathcal{Q}_{22} &= h^2 Q_{22} + G_{12} + G_{21}, \\
\mathcal{Q}_{2(n-1)} &= h^2 Q_{2(n-1)} + G_{2n} + G_{1(n-1)}, \\
\mathcal{Q}_{(n-1)2} &= h^2 Q_{(n-1)2} + G_{n2} + G_{(n-1)1}, \\
\mathcal{Q}_{(n-1)(n-1)} &= h^2 Q_{(n-1)(n-1)} + G_{n(n-1)} + G_{(n-1)n}, \\
\mathcal{Q}_{i2} &= h^2 Q_{i2} + G_{i1}, \\
\mathcal{Q}_{i(n-1)} &= h^2 Q_{i(n-1)} + G_{in}, \\
\mathcal{Q}_{2j} &= h^2 Q_{2j} + G_{1j}, \\
\mathcal{Q}_{(n-1)j} &= h^2 Q_{(n-1)j} + G_{nj},
\end{aligned}
$$

for $i, j = 3, \ldots, n-2$ and where again Q_{ij} are the F-transform components of q and G_{ij} came from the boundary condition by (4.5.7).

Matrix K_h has the following three-diagonal block structure

$$
K_h = \begin{bmatrix} H & -I & 0 & \ldots & 0 \\ -I & H & -I & \ldots & 0 \\ 0 & -I & I & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & -I & H & -I \\ 0 & \ldots & 0 & -I & H \end{bmatrix} \tag{4.5.9}
$$

where matrices H are square matrices of type $(n-2) \times (n-2)$ given by

$$
H = \begin{bmatrix} 4 & -1 & 0 & \ldots & 0 \\ -1 & 4 & -1 & \ldots & 0 \\ 0 & -1 & 4 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & -1 & 4 & -1 \\ 0 & \ldots & 0 & -1 & 4 \end{bmatrix}
$$

and matrices I are unit matrices of type $(n-2)\times(n-2)$ given as follows

$$I = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}.$$

Details relating to the algorithm can be found in [65]. Solution of system (4.5.3) is given by the following matrix

$$F^2[u] = \begin{bmatrix} U_{11} & U_{12} & \dots & U_{1m} \\ U_{21} & U_{22} & \dots & U_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots \\ U_{n1} & \dots & \dots & U_{nm} \end{bmatrix} \tag{4.5.10}$$

where the components U_{ij} are given either by (4.5.7) or they are determined as a solution to 4.5.8. This solution serves as a discrete numerical solution to (4.5.1) with boundary condition (4.5.2). To transfer it back to the space $C(\mathcal{D}^2)$, the inverse F-transform is applied.

4.6 Convergence of the Approximated Solutions

This section aims at proving important convergence properties of numerical solutions of equations of mathematical physics given by algorithms using the F-transform method and described in Section 4.3, Section 4.4 and finally in Section 4.5.

To make the result more transparent, let us again recall the notation that will be used below. We fix a uniform fuzzy partition $\{\mathbf{A}_1,\dots,\mathbf{A}_n\} \times_\odot \{\mathbf{B}_1,\dots,\mathbf{B}_m\}$ of $\mathcal{D}^2$ with equidistant steps h^x on X and h^y on Y. Any of the F-transforms below is computed with respect to this partition. Approximate solutions at nodes of the partition (c_i^x, c_j^y) are denoted by U_{ij} and analytical solutions u of the given partial differential equations at nodes of the partition (c_i^x, c_j^y) will be denoted as follows $u_{ij} = u(c_i^x, c_j^y)$. Analogously, the F-transform components of function q from the right-hand sides of the studied partial differential equations are denoted by Q_{ij} while their precise function values will be denoted as follows $q_{ij} = q(c_i^x, c_j^y)$.

Remark 14 *In the case of evolutionary equations, the domain is given as follows* $\mathcal{D}^2 = X \times T$*, the nodes of a given fuzzy partition are denoted by* (c_i^x, c_j^t) *and equidistant steps are given analogously by* h^x *on* X *and by* h^t *on* T*.*

4.6.1 Heat Equation

Here, we prove a convergence theorem that justifies our approach to an approximate solution to heat equation (4.3.1) from Section 4.3. Besides the convergence, this theorem confirms that the introduced numerical solution ia equal to the analytical one up to a certain accuracy. It uses results from the finite difference method [44] and the properties of the F-transform from Chapter 3 and Chapter 2 in consequence.

Let us recall, that in the case of the heat equation, domain $\mathcal{D}^2$ denotes the following rectangle

$$\mathcal{D}^2 = X \times T$$

where $X = [0, 1]$, $T = [0, R]$ and $R \in \mathbb{R}$ and $\mathcal{D}_0'^2 = (0, 1) \times (0, R]$.

Theorem 4 *Let* $u : X \times T \to \mathbb{R}$ *be a solution to equation (4.3.1)-(4.3.3). Assume that* u *a is four times continuously differentiable function with respect to* x *and twice continuously differentiable function with respect to* t*. Let us be given a uniform fuzzy partition* $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$ *of* $\mathcal{D}^2$ *such that* $0 < r \leq 1/2$ *where* $r = (\alpha h^t)/(h^x)^2$*. Let* $F^2[u]$ *be the approximate solution given by (4.3.13) . Then the norm of error of the approximate solution can be estimated as follows*

$$|| \, u_{ij} - U_{ij} \, || = O(h^t + (h^x)^2) \tag{4.6.1}$$

where the norm $\| \cdot \|$ is given as follows

$$\| u_{ij} - U_{ij} \| \equiv \max_{\substack{i=1,\dots,n \\ j=1,\dots,m}} | u_{ij} - U_{ij} | .$$

PROOF: Let us fix arbitrary $i = 2, \dots, n-1$ and $j = 2, \dots, m$. Since u is a solution to (4.3.1)-(4.3.3) on $\mathcal{D}_0'^2$, the following equality

$$\frac{\partial u(c_i^x, c_j^t)}{\partial t} - \alpha \frac{\partial^2 u(c_i^x, c_j^t)}{\partial x^2} = q(c_i^x, c_j^t), \quad \alpha \in \mathbb{R}^+ \tag{4.6.2}$$

is fulfilled for $i = 2, \dots, n-1$ and for $j = 2, \dots, m$. Finite differences keep the following accuracies

$$\frac{\partial u(c_i^x, c_j^t)}{\partial t} = \frac{u_{i(j+1)} - u_{ij}}{h^t} + O(h^t),$$

$$\frac{\partial^2 u(c_i^x, c_j^t)}{\partial x^2} = \frac{u_{(i-1)j} - 2u_{ij} + u_{(i+1)j}}{(h^x)^2} + O((h^x)^2)$$

of the first and second order, respectively. So, if we replace the partial derivatives on the left-hand side of (4.6.2) by the finite differences, the equation will be fulfilled up to accuracy $O(h^t + (h^x)^2)$ which means

$$\frac{u_{i(j+1)} - u_{ij}}{h^t} - \alpha \frac{u_{(i-1)j} - 2u_{ij} + u_{(i+1)j}}{(h^x)^2} = q(c_i^x, c_j^t) + O(h^t + (h^x)^2). \tag{4.6.3}$$

Therefore, we come to the conclucion that values u_{ij} fulfil equation (4.3.10) up to accuracy $h^t O((h^t) + (h^x)^2)$ which means

$$u_{i(j+1)} = r u_{(i-1)j} + (1 - 2r) u_{ij} + r u_{(i+1)j} + h^t q_{ij} + h^t O(h^t + (h^x)^2). \tag{4.6.4}$$

Due to Lemma 17, the F-transform components Q_{ij} approximate q_{ij}, precise values of q at the fuzzy partition nodes, up to $O((h^t)^2 + (h^x)^2)$.

Now, let us define an error of approximate solution at node (c_i^x, c_j^t) by

$$e_{ij} = | u_{ij} - U_{ij} | .$$

By using both previous estimations we can continue as follows

$$e_{i(j+1)} = re_{(i-1)j} + (1-2r)e_{ij} + re_{(i+1)j} + h^t O((h^t)^2 + (h^x)^2) + h^t O(h^t + (h^x)^2)$$

which means

$$e_{i(j+1)} = re_{(i-1)j} + (1-2r)e_{ij} + re_{(i+1)j} + h^t O(h^t + (h^x)^2). \tag{4.6.5}$$

From initial condition (4.3.2) and boundary conditions (4.3.3) we get $e_{i1} = 0$, for $i = 1, \ldots, n$ and $e_{1j} = e_{nj} = 0$ for $j = 1, \ldots, m$.

If we consider the following column norm

$$\| e_j \|_c = \max_{i=1,\ldots,n} | e_{ij} |,$$

we obtain $\| e_1 \|_c = 0$, which is going to be used at the end of the proof.

Since $0 < r \leq 1/2$, all the coefficients on the right-hand side of equation (4.6.5) are non-negative and their sum is equal to 1. Then

$$\begin{aligned} | e_{i(j+1)} | &= r \, | e_{(i-1)j} | + (1-2r) \, | e_{ij} | + r \, | e_{(i+1)j} | \\ &\quad + h^t O(h^t + (h^x)^2) \leq \| e_j \|_c + h^t O(h^t + (h^x)^2) \end{aligned}$$

which yields

$$\| e_{j+1} \|_c \leq \| e_j \|_c + h^t O(h^t + (h^x)^2).$$

This implies

$$\begin{aligned} \| e_j \|_c &\leq \| e_{j-1} \|_c + h^t O(h^t + (h^x)^2) \\ \| e_j \|_c &\leq \| e_{j-2} \|_c + 2h^t O(h^t + (h^x)^2) \\ &\ldots \\ \| e_j \|_c &\leq \| e_1 \|_c + (j-1)h^t O(h^t + (h^x)^2) \end{aligned}$$

and with the help of the proved equality $\| e_1 \|_c = 0$ and the fact that $(j-1)h^t$ is a finite number less than R, we can write

$$\| e_j \|_c = O(h^t + (h^x)^2)$$

for an arbitrary j which proves the theorem. $\square$

Now, we prove a convergence of the inverse F-transform of the numerical solution $F^2[u]$ to the precise analytical solution to equations (4.3.1)-(4.3.3).

Corollary 6 *Let all the assumptions of Theorem 4 be fulfilled and furthermore, let* $\{\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}\}_{n,m}$ *be a sequence of uniform fuzzy partitions of* $\mathcal{D}^2$*. Let* $\{u^F_{n,m}\}_{n,m}$ *be a sequence of the inverse F-transforms applied to the approximate solutions* $F^2[u]$ *given by (4.3.13), one for each fuzzy partition. Then the sequence* $\{u^F_{n,m}\}_{n,m}$ *uniformly converges to the analytical solution* u *of equation (4.3.1)-(4.3.3) on* $\mathcal{D}^2$*.*

PROOF: Since u is continuous on $\mathcal{D}^2$, for each $\varepsilon > 0$ there exists $\delta > 0$ such that for all $(x_1, t_1), (x_2, t_2) \in \mathcal{D}^2$:

$$\| (x_1, t_1) - (x_2, t_2) \|_{\epsilon} < \delta \text{ implies } \mid u(x_1, t_1) - u(x_2, t_2) \mid < \varepsilon$$

where $\| \cdot \|_{\epsilon}$ denotes the Euclidean norm.

Let us fix some $\varepsilon > 0$. Let $\{\mathbf{A}_1, \ldots, \mathbf{A}_{n_\varepsilon}\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_{m_\varepsilon}\}$ be a uniform fuzzy partition such that $h^x < \delta/\sqrt{2}$ and $h^t < \delta/\sqrt{2}$.

Furthermore, we choose arbitrary $(z_1, z_2) \in [c^x_i, c^x_{i+1}] \times [c^t_j, c^t_{j+1}]$ where $2 \le i \le n_\varepsilon - 2$ and $2 \le j \le m_\varepsilon - 2$.

Because $h^x < \delta/\sqrt{2}$ and $h^t < \delta/\sqrt{2}$ we obtain

$$\| (x_1, t_1) - (x_2, t_2) \|_{\epsilon} < \sqrt{\frac{\delta^2}{2} + \frac{\delta^2}{2}} = \delta$$

which due to the continuity implies

$$| u(z_1, z_2) - u(c_i^x, c_j^t) | < \varepsilon.$$

The absolute difference $| u(z_1, z_2) - U_{ij} |$ can be estimated as follows

$$| u(z_1, z_2) - U_{ij} | \leq | u(z_1, z_2) - u(c_i^x, c_j^t) | + | u(c_i^x, c_j^t) - U_{ij} | = \varepsilon + O(h^t + (h^x)^2).$$

Analogously, way we obtain

$$| u(z_1, z_2) - U_{(i+1)j} | \leq \varepsilon + O(h^t + (h^x)^2),$$

$$| u(z_1, z_2) - U_{i(j+1)} | \leq \varepsilon + O(h^t + (h^x)^2),$$

$$| u(z_1, z_2) - U_{(i+1)(j+1)} | \leq \varepsilon + O(h^t + (h^x)^2).$$

Finally, we can estimate the following difference

$$| u(z_1, z_2) - u^F_{n_\varepsilon, m_\varepsilon}(z_1, z_2) | = | u(z_1, z_2) - \sum_{i=1}^{n_\varepsilon} \sum_{j=1}^{m_\varepsilon} \mathbf{A}_i(z_1) \mathbf{B}_j(z_2) U_{ij} |$$

which is less or equal to

$$\sum_{i=1}^{n_\varepsilon} \sum_{j=1}^{m_\varepsilon} \mathbf{A}_i(z_1) \mathbf{B}_j(z_2) | u(z_1, z_2) - U_{ij} |$$

which is less or equal to

$$\sum_{i=1}^{n_\varepsilon} \sum_{j=1}^{m_\varepsilon} \mathbf{A}_i(z_1) \mathbf{B}_j(z_2) (\varepsilon + O(h^t + (h^x)^2))$$

and therefore we can state

$$| u(z_1, z_2) - u^F_{n_\varepsilon, m_\varepsilon}(z_1, z_2) | < \varepsilon + O(h^t + (h^x)^2)$$

Therefore, the inverse F-transform computed from the components determined by algorithm (4.3.10) uniformly converges to the precise (analytical) solution.

$\square$

4.6.2 Wave Equation

Here, we prove the convergence theorem which justifies our approach to an approximate solution to wave equation (4.4.1) from Section 4.4. Besides the convergence, this theorem confirms that the introduced numerical solution is equal to the analytical one up to a certain accuracy. It uses results from the finite difference method [44] and the properties of the F-transform from Chapter 3 and Chapter 2 in consequence.

Let us recall, that in the case of the wave equation, domain $\mathcal{D}^2$ denotes the following rectangle

$$\mathcal{D}^2 = X \times T$$

where $X = [0, 1]$, $T = [0, R]$ and $R \in \mathbb{R}$ and $\mathcal{D}_0'^2 = (0, 1) \times (0, R]$.

Theorem 5 *Let $u : X \times T \to \mathbb{R}$ be a solution to equation (4.4.1)-(4.4.3). Assume that u is a four times continuously differentiable function with respect to x and a four time continuously differentiable function with respect to t. Let us be given a uniform fuzzy partition $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_\odot \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$ of $\mathcal{D}^2$ such that $0 < r \le 1$ where $r = h^t/(\alpha h^x)$. Let $F^2[u]$ be the approximate solution given by (4.4.11). Then the norm of error of the approximate solution can be estimated as follows*

$$|| u_{ij} - U_{ij} || = O((h^t)^2 + (h^x)^2) \tag{4.6.6}$$

where the norm $\| \cdot \|$ is given as follows

$$|| u_{ij} - U_{ij} || \equiv \max_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} | u_{ij} - U_{ij} | .$$

PROOF: Let us fix arbitrary $i = 2, \ldots, n-1$ and $j = 2, \ldots, m$. Since u is a solution to (4.4.1)-(4.4.3) on $\mathcal{D}_0'^2$, the following equality

$$\frac{\partial^2 u(c_i^x, c_j^t)}{\partial x^2} - \alpha^2 \frac{\partial^2 u(c_i^x, c_j^t)}{\partial t^2} = q(c_i^x, c_j^t), \quad \alpha \in \mathbb{R} \tag{4.6.7}$$

is fulfilled for $i = 2, \ldots, n-1$ and for $j = 2, \ldots, m$. Finite differences keep the following accuracies of the second order

$$\frac{\partial^2 u(c_i^x, c_j^t)}{\partial x^2} = \frac{u_{(i-1)j} - 2u_{ij} + u_{(i+1)j}}{(h^x)^2} + O((h^x)^2),$$

$$\frac{\partial^2 u(c_i^x, c_j^t)}{\partial t^2} = \frac{u_{i(j-1)} - 2u_{ij} + u_{i(j+1)}}{(h^t)^2} + O((h^t)^2),$$

respectively. So, if we replace the partial derivatives on the left hand-side of (4.6.7) by the finite differences, the equation will be fulfilled up to accuracy $O((h^t)^2 + (h^x)^2)$ which means

$$\frac{u_{(i-1)j} - 2u_{ij} + u_{(i+1)j}}{(h^x)^2} - \alpha^2 \frac{u_{i(j-1)} - 2u_{ij} + u_{i(j+1)}}{(h^t)^2} = q_{ij} + O((h^t)^2 + (h^x)^2).$$

Therefore, we come to the fact that values u_{ij} fulfil equation (4.4.6) up to accuracy $(h^t)^2 O((h^t) + (h^x)^2)$ which means that

$$u_{i(j+1)} = r^2 u_{(i-1)j} + 2(1-r^2)u_{ij} + r^2 u_{(i+1)j} - u_{i(j-1)} - (h^t)^2 q_{ij} + (h^t)^2 O((h^t)^2 + (h^x)^2).$$

Due to Lemma 17, the F-transform components Q_{ij} approximate values q_{ij} up to $O((h^t)^2 + (h^x)^2)$.

Let the error e_{ij} of the approximate solution at node (c_i^x, c_j^t) be defined as follows

$$e_{ij} = |u_{ij} - U_{ij}|.$$

Using both previous estimations we continue as follows

$$e_{i(j+1)} = r^2 e_{(i-1)j} + 2(1-r^2)e_{ij} + r^2 e_{(i+1)j} - e_{i(j-1)} + (h^t)^2 O((h^t)^2 + (h^x)^2).$$

From initial conditions (4.4.2) and boundary conditions (4.4.3) we get $e_{i1} = 0$, for $i = 1, \ldots, n$ and $e_{1j} = e_{nj} = 0$ for $j = 1, \ldots, m$.

Again, let the column norm $||\cdot||_c$ be given as follows

$$|| e_j ||_c = \max_{i=1,\ldots,n} | e_{ij} | .$$

Obviously, $\| e_1 \|_c = 0$.

We can continue as follows

$$
\begin{aligned}
|e_{i(j+1)}| &= r^2|e_{(i-1)j}| + 2(1-r^2)|e_{ij}| + r^2|e_{(i+1)j}| - |e_{i(j-1)}| \\
&\quad + (h^t)^2 O((h^t)^2 + (h^x)^2) \\
|e_{i(j+1)}| &\leq \| e_j \|_c - |e_{i(j-1)}| + (h^t)^2 O((h^t)^2 + (h^x)^2)
\end{aligned}
$$

which yields

$$\| e_{j+1} \|_c \leq \| e_j \|_c + (h^t)^2 O((h^t)^2 + (h^x)^2).$$

This implies

$$
\begin{aligned}
&\| e_j \|_c \leq \| e_{j-1} \|_c + (h^t)^2 O((h^t)^2 + (h^x)^2) \\
&\| e_j \|_c \leq \| e_{j-2} \|_c + 2(h^t)^2 O((h^t)^2 + (h^x)^2) \\
&\qquad\qquad \dots \\
&\| e_j \|_c \leq \| e_1 \|_c + (j-1)(h^t)^2 O((h^t)^2 + (h^x)^2)
\end{aligned}
$$

and with help of $\| e_1 \|_c = 0$ and the fact that $(j-1)(h^t)^2$ is a finite number less than R^2, we can write

$$\| e_j \|_c = O((h^t)^2 + (h^x)^2)$$

for an arbitrary j which proves the theorem.

□

Now, we prove a convergence of the inverse F-transform of the numerical solution $F^2[u]$ to the precise analytical solution to equations (4.4.1)-(4.4.3).

Corollary 7 *Let all the assumptions of Theorem 5 be fulfilled and furthermore, let* $\{\{\mathbf{A}_1, \dots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \dots, \mathbf{B}_m\}\}_{n,m}$ *be a sequence of uniform fuzzy partitions of* $\mathcal{D}^2$*. Let* $\{u^F_{n,m}\}_{n,m}$ *be a sequence of the inverse F-transforms applied to the approximate* *solutions $F^2[u]$ given by (4.4.11), one for each fuzzy partition. Then the sequence $\{u^F_{n,m}\}_{n,m}$ uniformly converges to the analytical solution u of equation (4.4.1)-(4.4.3) on $\mathcal{D}^2$.*

PROOF: Proof of the corollary is analogous to the proof of Corollary 6 and therefore it is omitted. □

4.6.3 Poisson's Equation

This subsection, analogously to both previous subsections, discusses convergence and accuracy of the proposed numerical solution.

Let us recall, that in the case of Poisson's equation, domain $\mathcal{D}^2$ denotes the following rectangle

$$\mathcal{D}^2 = X \times Y$$

where $X = [0,1]$, $Y = [0,1]$ and $\mathcal{D}'^2_0 = (0,1) \times (0,1)$.

Theorem 6 *Let $u : X \times T \to \mathbb{R}$ be a solution to equation (4.5.1)-(4.5.2). Assume that u is a four times continuously differentiable function with respect to x and a four times continuously differentiable function with respect to t. Let us be given a uniform fuzzy partition $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_\odot \{\mathbf{B}_1, \ldots, \mathbf{B}_n\}$ of $\mathcal{D}^2$. Let $F^2[u]$ be the approximate solution given by (4.5.10). Then the norm of error of the approximate solution can be estimated as follows*

$$|| \, u_{ij} - U_{ij} \, || = O(h^2) \tag{4.6.8}$$

where the norm $\| \cdot \|$ is given as follows

$$|| \, u_{ij} - U_{ij} \, || \equiv \max_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}} | \, u_{ij} - U_{ij} \, | \, .$$

PROOF: Let us fix arbitrary $i, j = 2, \ldots, n-1$. Since u is a solution to (4.5.1)-(4.5.2) on $\mathcal{D}_0^2$, the following equality

$$\frac{\partial^2 u(c_i^x, c_j^t)}{\partial t^2} + \frac{\partial^2 u(c_i^x, c_j^t)}{\partial x^2} = -q(c_i^x, c_j^t) \tag{4.6.9}$$

is fulfilled. Finite differences keep the accuracies of the second order which yields

$$\frac{1}{h^2}(4u_{ij} - u_{(i-1)j} - u_{(i+1)j} - u_{i(j-1)} - u_{i(j+1)}) + O(h^2) = q_{ij} \tag{4.6.10}$$

for $i, j = 2, \ldots, n-1$.

So, we can write

$$K_h u_h^p = q_h^p + O_{h^2} \tag{4.6.11}$$

where matrix K_h is given by (4.5.9) and where u_h^p and q_h^p are the following real vectors

$$u_h^p = [u_{22}, \ldots, u_{2(n-1)}; u_{32}, \ldots, u_{3(n-1)}; \ldots; u_{(n-1)2}, \ldots, u_{(n-1)(n-1)}]^T$$

and

$$q_h^p = [\mathcal{Q}_{22}^p, \ldots, \mathcal{Q}_{2(n-1)}^p; \mathcal{Q}_{32}^p, \ldots, \mathcal{Q}_{3(n-1)}^p; \ldots; \mathcal{Q}_{(n-1)2}^p, \ldots, \mathcal{Q}_{(n-1)(n-1)}^p]^T,$$

respectively and where

$$\begin{aligned}
\mathcal{Q}_{ij}^p &= h^2 q_{ij}, \\
\mathcal{Q}_{22}^p &= h^2 q_{22} + G_{12} + G_{21}, \\
\mathcal{Q}_{2(n-1)}^p &= h^2 q_{2(n-1)} + G_{2n} + G_{1(n-1)}, \\
\mathcal{Q}_{(n-1)2}^p &= h^2 q_{(n-1)2} + G_{n2} + G_{(n-1)1}, \\
\mathcal{Q}_{(n-1)(n-1)}^p &= h^2 q_{(n-1)(n-1)} + G_{n(n-1)} + G_{(n-1)n}, \\
\mathcal{Q}_{i2}^p &= h^2 q_{i2} + G_{i1}, \\
\mathcal{Q}_{i(n-1)}^p &= h^2 q_{i(n-1)} + G_{in}, \\
\mathcal{Q}_{2j}^p &= h^2 q_{2j} + G_{1j}, \\
\mathcal{Q}_{(n-1)j}^p &= h^2 q_{(n-1)j} + G_{nj},
\end{aligned}$$

for $i, j = 3, \ldots, n-2$. Finally, O_{h^2} is the following $(n-2)\cdot(n-2)$-dimensional vector

$$O_{h^2} = [O(h^2), \ldots, O(h^2)]^T.$$

When we subtract equation (4.5.8) from equation (4.6.11), we obtain

$$K_h e_h = q_h - Q_h + O_{h^2} \tag{4.6.12}$$

where $e_h = (u_h - U_h)$.

Since $q_{ij} - Q_{ij} = O(h^2)$ we come to

$$K_h e_h = h^2 O_{h^2} \tag{4.6.13}$$

which means that $||e_{ij}|| = O(h^2)$. □

Now, let us prove the convergence of the inverse F-transform of the numerical solution U_{ij} to the precise solution of (4.4.1)-(4.4.3).

Corollary 8 *Let all the assumptions of Theorem 6 be fulfilled and furthermore, let $\{\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_n\}\}_n$ be a sequence of uniform fuzzy partitions of $\mathcal{D}^2$. Let $\{u^F_{n,n}\}_n$ be a sequence of the inverse F-transforms applied to the approximate solutions $F^2[u]$ given by (4.5.10), one for each fuzzy partition. Then the sequence $\{u^F_{n,n}\}_n$ uniformly converges to the analytical solution u of equation (4.5.1)-(4.5.2) on $\mathcal{D}^2$.*

PROOF: Proof of the corollary is analogous to the proof of Corollary 6 and therefore it is omitted. □

4.7 Demonstration

In physics, an integral usually means an energy, e.g., heat. For a demonstration let us consider the heat equation where the solution $u(x,t)$ means a temperature at point x in time t. The convergence property is required but it is not sufficient because, e.g., in case of convergence of a numerical solution to the analytical one from below (or alternatively from above), the computed heat flow would be too far from the real heat flow, although the temperatures would be quite close to each other. Therefore the integral proximity is found to be an important quality indicator which advocates in favour of the F-transform method.

Let us demonstrate the advantage of the numerical solution based on the F-transform in comparison with the ordinary numerical solution using the finite difference method. Let $f : \mathcal{D}^2 \to \mathbb{R}$ be an arbitrary continuous function on a given domain $\mathcal{D}^2$ and let $\{\mathbf{A}_1, \ldots, \mathbf{A}_n\} \times_{\odot} \{\mathbf{B}_1, \ldots, \mathbf{B}_m\}$ be a fuzzy partition of $\mathcal{D}^2$. Then the following piecewise integral least square criterion

$$\Psi([c_{ij}]) = \int_c^d \int_a^b \sum_{i=1}^{n} \sum_{j=1}^{m} (f(x,t) - c_{ij})^2 \mathbf{A}_i(x) \mathbf{B}_j(t) dx dt \tag{4.7.1}$$

is minimized by the F-transform component matrix $[F_{ij}]$ given by (3.1.2).

Criterion (4.7.1) characterizes an integral proximity. The uniform convergence in $C(\mathcal{D}^2)$ in combination with a minimization of the previous integral criterion provides a powerful tool that will be explained.

When the right-hand side $q(x,t)$ of equation (4.3.1) is damaged by a noise (inaccuracies in measurements), the ordinary finite difference method generates a damaged approximate solution. For more results related to the noise removing by the F-transform we refer to [61].

Let us consider the following example. Let $\mathcal{D}^2 = [0,1] \times [0,1]$ and let

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 5(-4(x-1/2)^2 + 1)e^{-t/2}, \quad (x,t) \in \mathcal{D}_0'^2, \tag{4.7.2}$$

furthermore let us be given the following boundary conditions

$$u(0,t) = 0, \quad u(1,t) = 0, \quad t \in [0,1] \tag{4.7.3}$$

and the following initial condition

$$u(x,0) = \begin{cases} 2x, & 0 \le x \le \frac{1}{2}, \\ 2(1-x), & \frac{1}{2} \le x \le 1. \end{cases} \tag{4.7.4}$$

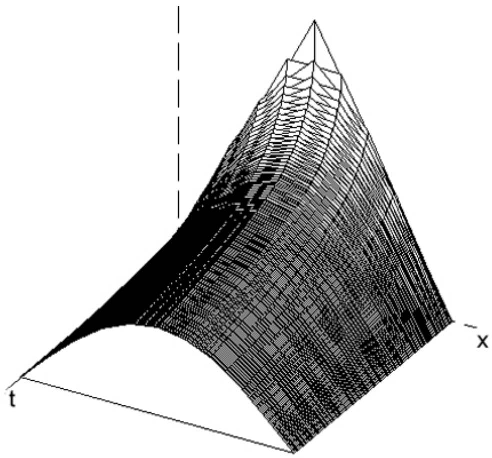
Equation (4.7.2)-(4.7.4) describes a distribution of temperature u of a metal rod at point x and time t while both ends of this rod are kept in contact with melting ice. The right-hand side $5(-4(x-1/2)^2+1)e^{-t/2}$ gives us information about heat sources. The solution of equation (4.7.2) is displayed on Figure 4.1(a).

Let us consider a situation (very common in practice) when the right-hand side of equation (4.7.2) is obtained by some measurements with inaccuracies. These inaccuracies can be modelled by adding random noise.

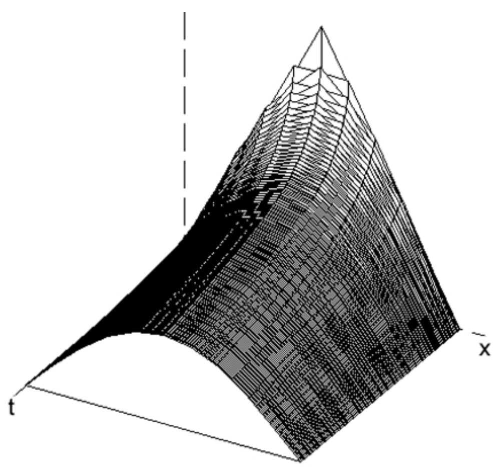
For the demonstration, we have added random noise with Gaussian distribution with 0 mean and the standard deviation equal to 2, to the right-hand side of equation (4.7.2). Such a modified equation has been numerically solved by both methods: the finite difference method and by the F-transform technique. In both cases we have used $n = 11$ nodes on axis x and $m = 200$ nodes on the time axis.

The difference is significant. The solution using the F-transform is practically the same as in the case without any noise, see Figure 4.1(a) and Figure 4.1(b). On the other hand, the solution based on the ordinary technique without any help of the F-transform is highly influenced by the noise, see Figure 4.1(c).

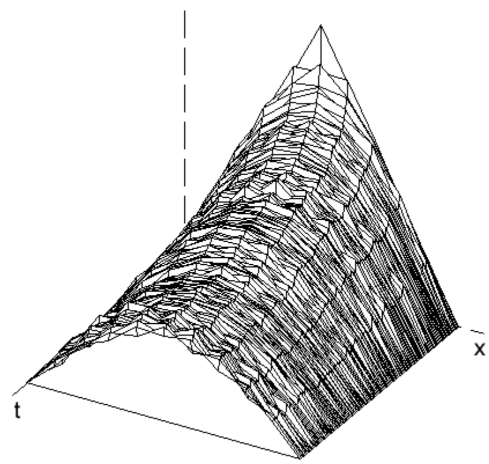
This advocates in favour of the F-transform technique.



(a) Solution of equation (4.7.2)



(b) Numerical solution of modified equation (4.7.2). Finite difference method with help of F-transform.



(c) Numerical solution of modified equation (4.7.2). Finite difference method.

Figure 4.1: An illustration of the difference between numerical solutions with and without the F-transform

Chapter 5

Normal Forms and their connection to the F-transform

This chapter has been motivated by two, up to now unrelated, approaches belonging to *fuzzy approximation* framework. As mentioned in Section 1.3, fuzzy approximation is understood as a collection of techniques for an approximation of functional dependencies between variables by means of the fuzzy set theory [9, 32, 52].

Two main approaches recalled already in Section 1.3 follow a similar task but use different tools and methods. The first one, which relates to the so-called *normal forms* [13, 56], consists in an investigation of different interpretations of fuzzy rules. Two standard interpretations [21, 46, 49] lead to two standard normal forms - *disjunctive and conjunctive normal forms* - formalizing these interpretations. The second approach represented typically by the F-transform technique aims at an approximation of a given function by another function preserving the feature of transparent interpretability. The approximated functional dependency is expressed as a linear combination of basic functions where basic functions represent local subdomains.

The main goal of the chapter is to investigate normal forms and their approximation abilities and interrelate them with the F-transform. We will also show that F-transform fits even into the framework of the normal forms. This will require the development of a formalization of the additive interpretations [27] of fuzzy systems. The formalization is based on the *additive normal form* from [59] which is furthermore generalized and investigated in Section 5.2.

We achieve a generalized class of fuzzy transforms based on an arbitrary strict continuous t-norm. The relationship to the normal forms allows us to use their formal apparatus to inherit some particular results, especially an upper bound of an approximation error.

5.1 Extensionality Property

Throughout the entire chapter, we assume that $\mathcal{D}$ is a nonempty set of objects and $\mathbf{F}$ is a fuzzy set on $\mathcal{D}$. We denote this fact by $\mathbf{F} \subseteq\!\!\!\!\sim \mathcal{D}$.

Remark 15 *For the domain we will again use symbol $\mathcal{D}$ used in the previous chapters but because normal forms do not have to be defined on closed intervals or cubes (products of intervals) $\mathcal{D}$ denotes generally a non-empty set of objects in this chapter.*

Extensionality is a well-known notion from the classical set theory. A generalized version of this notion has been introduced in [36]. There, extensional fuzzy sets (relations) are defined with respect to a *similarity* relation on their domain where the notion similarity is defined as follows.

Definition 32 Let $*$ be a t-norm and $\mathbf{S} \subseteq\!\!\!\!\sim \mathcal{D} \times \mathcal{D}$. Then $\mathbf{S}$ is called

1. *reflexive* if $\mathbf{S}(x, x) = 1$,

2. *symmetric* if $\mathbf{S}(x,y) = \mathbf{S}(y,x)$,

3. $*$-*transitive* if $\mathbf{S}(x,y) * \mathbf{S}(y,z) \leq \mathbf{S}(x,z)$,

for all $x, y, z \in \mathcal{D}$. If $\mathbf{S}$ is reflexive, symmetric and $*$-transitive then we say that $\mathbf{S}$ is a $*$-*similarity* relation.

The similarity relation expresses a natural formalization of, say closeness or proximity, i.e., equality up to some degree. If we fix a node $c \in \mathcal{D}$ then a similarity $\mathbf{S}(c,y)$ is a symmetric fuzzy number on $\mathcal{D}$ which represents a neighborhood of the node c or local sub-domain of $\mathcal{D}$ around c.

Fuzzy rules very often use symmetric fuzzy numbers as antecedent fuzzy sets, therefore a similarity fuzzy relation $\mathbf{S}$ is used in normal forms, see Section 1.3.1. But this is too restrictive and the discussed fuzzy relation may be handled more generally.

Therefore, we present a more general case of extensionality then the ones published in [36]. The reason comes from the fact that extensional fuzzy sets defined with respect to a similarity have properties relating to Lipschitz continuity. Let us recall the paper [43] where it was proved that in a t-norm based algebra, the extensionality of a fuzzy set with respect to a similarity is equivalent to Lipschitz continuity with respect to the pseudo-metric induced by the given similarity. An analogous result was published in [10] even for fuzzy relations which are not similarity relations.

Definition 33 [10] Let $\mathbf{F} \underset{\sim}{\subset} \mathcal{D}$, $\mathbf{E} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}$, and $*$ be a t-norm. We say that $\mathbf{F}$ is *extensional* with respect to $\mathbf{E}$ and $*$ on $\mathcal{D}$ if for each $x, y \in \mathcal{D}$:

$$\mathbf{E}(x,y) * \mathbf{F}(x) \leq \mathbf{F}(y). \tag{5.1.1}$$

Remark 16 *If* $\mathbf{F} \underset{\sim}{\subseteq} \mathcal{D}^n$ *then* $\mathbf{E} \underset{\sim}{\subseteq} \mathcal{D}^n \times \mathcal{D}^n$ *is given as a combination of* n *binary fuzzy relations on* $\mathcal{D}$ *i.e.*

$$\mathbf{E}(\bar{x}, \bar{y}) = \mathbf{E}_1(x_1, y_1) *_1 \cdots *_{n-1} \mathbf{E}_n(x_n, y_n),$$

where $*_1, \ldots, *_{n-1}$ *are arbitrary t-norms,* $\mathbf{E}_i \underset{\sim}{\subseteq} \mathcal{D} \times \mathcal{D}$ *for* $i = 1, \ldots, n$ *and* $\bar{x} = (x_1, \ldots, x_n), \bar{y} = (y_1, \ldots y_n) \in \mathcal{D}^n$*.*

Assuming $*_1 = \ldots = *_{n-1} = *$*, we come to well-known form of the extensionality see e.g. [13].*

5.2 Normal Forms and Their Approximation Abilities

Normal forms are presented in this section especially with the aim to have an approximation of a fuzzy set with an arbitrary precision. Information about the error of approximation is contained in an inequality called conditional equivalence.

First of all, let us recall a remark from [38] concerning the extension of t-conorms to infinitary operations.

Remark 17 *Each t-conorm* $\sqcup$ *can be extended to a (countably) infinitary operation putting for each* $(x_i)_{i \in \mathbb{N}} \in [0,1]^{\mathbb{N}}$ *i.e. for each sequence* $(x_i)_{i \in \mathbb{N}}$ *such that* $x_i \in [0,1]$ *for all* $i \in \mathbb{N}$*:*

$$\bigsqcup_{i}^{\infty} x_i = \lim_{n \to \infty} \bigsqcup_{i}^{n} x_i. \tag{5.2.1}$$

Note that the limit on the right-hand side always exists since the sequence

$$\left(\bigsqcup_{i}^{n} x_i \right)_{i \in \mathbb{N}}$$

is non-decreasing and bounded from above.

We only mention that for an arbitrary (not necessarily countable) index set I and $(x_i)_{i\in\mathbb{N}} \in [0,1]^I$, i.e., a family $(x_i)_{i\in I}$ with $x_i \in [0,1]$ for all $i \in I$, the following

$$\bigsqcup_{i\in I} x_i = \bigvee \left\{ \bigsqcup_{j=1}^{k} \mid (x_{i_1}, x_{i_2}, \ldots, x_{i_k}) \text{ is finite subfamily of } (x_i)_{i\in I} \right\} \tag{5.2.2}$$

is a well-defined formula.

Remind that, we assume that $\mathbf{F} \underset{\sim}{\subset} \mathcal{D}$, $\mathbf{E} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}$, $*$ be a left-continuous t-norm and $\rightarrow_*$ its residuation and finally, $\oplus$ is Łukasiwicz t-conorm. Moreover, let us fix a set of nodes $\mathcal{N} \subseteq \mathcal{D}$.

Definition 34 [14] The following expressions on the right-hand sides are the *disjunctive, conjunctive* and *additive* normal form of $\mathbf{F}$ with respect to $\mathcal{N}$

$$\mathbf{F}_{DNF,*}(x) = \bigvee_{c\in\mathcal{N}} (\mathbf{E}(c,x) * \mathbf{F}(c)), \tag{5.2.3}$$

$$\mathbf{F}_{CNF,*}(x) = \bigwedge_{c\in\mathcal{N}} (\mathbf{E}(x,c) \rightarrow_* \mathbf{F}(c)), \tag{5.2.4}$$

$$\mathbf{F}_{ANF,*}(x) = \bigoplus_{c\in\mathcal{N}} (\mathbf{E}(c,x) * \mathbf{F}(c)), \tag{5.2.5}$$

respectively.

For the definition of $\oplus$ over the infinite index set we refer to Remark 17 taken from [38]. Definitions of $\mathbf{F}_{DNF,*}, \mathbf{F}_{CNF,*}$ are taken from [13] and $\mathbf{F}_{ANF,*}$ is generalized version of $\mathbf{F}_{ANF,\odot}$ from [59]. If there will be a danger of confusion, we will add the index $\mathcal{N}$ to highlight the difference between particular normal forms.

The following definition of the orthogonality is taken from [59].

Definition 35 We say that $\mathbf{E} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}$ fulfils the *orthogonality* property with respect to $\mathcal{N} \subseteq \mathcal{D}$, if

$$\bigoplus_{c\in\mathcal{N}\setminus\{d\}} \mathbf{E}(c,x) = 1 - \mathbf{E}(d,x), \tag{5.2.6}$$

is valid for each $x \in \mathcal{D}$ and $d \in \mathcal{N}$.

Obviously, if $\mathcal{N}_1 \subseteq \mathcal{N}_2$ then $\mathbf{F}^{\mathcal{N}_1}_{DNF,*}(x) \leq \mathbf{F}^{\mathcal{N}_2}_{DNF,*}(x)$, $\mathbf{F}^{\mathcal{N}_1}_{ANF,*}(x) \leq \mathbf{F}^{\mathcal{N}_2}_{ANF,*}(x)$, and $\mathbf{F}^{\mathcal{N}_2}_{CNF,*}(x) \leq \mathbf{F}^{\mathcal{N}_1}_{CNF,*}(x)$. As we will see, the disjunctive and conjunctive normal forms of an extensional fuzzy set $\mathbf{F}$ give lower and upper approximations of $\mathbf{F}$, respectively [13].

Remark 18 *It is well-known, that in Boolean algebra of functions a normal forms represents an object of the algebra. Thus normal form is a special kind of representations of objects. If we generalize to say BL-alegbras [13, 59] we lose this property of representation. What we can get is an approximate representation ability.*

It should be stressed that normal forms are syntactical formulas i.e. expressions. So, a normal form is no function and no normal form can approximate anything. However, $\mathbf{F} \underset{\sim}{\subset} \mathcal{D}$ can be expressed in a normal form and we will talk about normal form of $\mathbf{F}$ as in Definition 34. And a normal form of $\mathbf{F}$ is already a fuzzy set (i.e. a function). Therefore we can claim that a normal for of $\mathbf{F}$ approximates $\mathbf{F}$.

Proposition 1 *If $\mathbf{F} \underset{\sim}{\subset} \mathcal{D}$ is extensional with respect to $\mathbf{E} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}$ and $*$ then*

$$\mathbf{F}_{DNF,*}(x) \leq \mathbf{F}(x) \leq \mathbf{F}_{CNF,*}(x), \tag{5.2.7}$$

for all $x \in \mathcal{D}$.

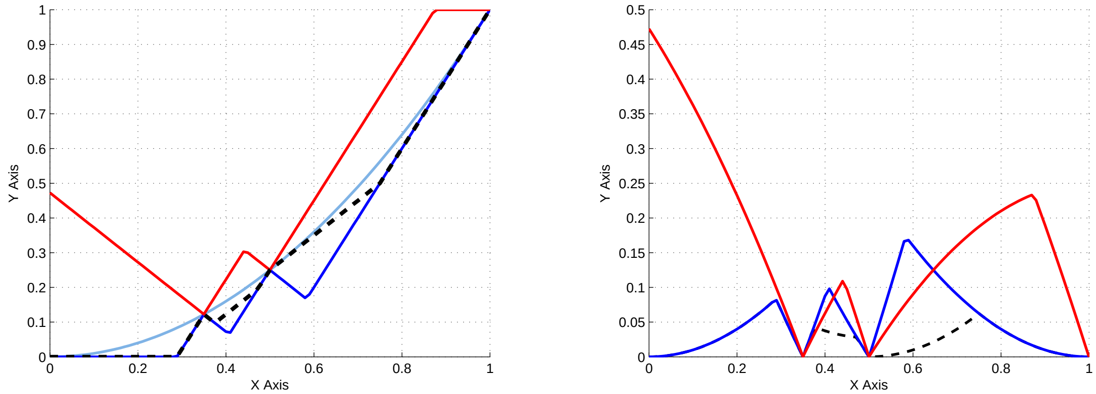
Let us illustrate the relationships between normal forms on the following example.

Example 3 *Let us consider the following one-dimensional case where the approximated fuzzy set*

$$\mathbf{F}(x) = x^2$$

is defined on $\mathcal{D} = [0, 1]$. Binary fuzzy relation $\mathbf{E}$ is given by

$$\mathbf{E}(x, y) = (x \leftrightarrow_{\otimes} y)^2 \otimes (y \leftrightarrow_{\otimes} x),$$



(a) Relationship between normal forms

(b) Error of approximation

Figure 5.1: An illustration of approximation abilities of all three given normal forms for the fuzzy set from Example 3. The dashed black line represents $\mathbf{F}_{ANF,\otimes}$, the red line depicts $\mathbf{F}_{CNF,\otimes}$ and the blue line belongs to $\mathbf{F}_{DNF,\otimes}$.

and the set of nodes $\mathcal{N} = \{0.35, 0.5, 1\}$. Finally, let $*$ be the Łukasiweicz t-norm $\otimes$. Let us mention that the given $\mathbf{F}$ is extensional w.r.t. the given $\mathbf{E}$ and $\otimes$, see [10].

Next, we obtain a relationship between the conjunctive, disjunctive and additive normal forms which is illustrated on Figure 5.1.

Obviously, without any assumption on $\mathbf{F}$ or $\mathbf{E}$, the inequality $\mathbf{F}_{DNF,*}(x) \leq \mathbf{F}_{ANF,*}(x)$ is valid for each $x \in \mathcal{D}$. Based on the result of I. Perfilieva in [59], the following proposition holds.

Proposition 2 *[14] Let $\mathbf{F} \underset{\sim}{\subset} \mathcal{D}$ and $\mathbf{E} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}$. If $\mathbf{E}$ is symmetric and fulfils the orthogonality condition with respect to $\mathcal{N}$ then for all $x \in \mathcal{D}$*

$$\mathbf{F}_{DNF,*}(x) \leq \mathbf{F}_{ANF,*}(x) \leq \mathbf{F}_{CNF,\otimes}(x). \tag{5.2.8}$$

PROOF: The proof uses a technique proposed in [59]. Let us recall the technique in short. The first inequality

$$\mathbf{F}_{DNF,*}(x) \leq \mathbf{F}_{ANF,*}(x)$$

is obvious since $\vee$ is the least t-conorm. The second part is proved by the following sequence of inequalities.

$$\mathbf{E}(c,x) * \mathbf{F}(c) \leq \mathbf{E}(c,x)$$
$$\bigoplus_{c \in \mathcal{N} \setminus \{d\}} (\mathbf{E}(c,x) * \mathbf{F}(c)) \leq \bigoplus_{c \in \mathcal{N} \setminus \{d\}} \mathbf{E}(c,x)$$

due to orthogonality of $\mathbf{E}$

$$\bigoplus_{c \in \mathcal{N} \setminus \{d\}} (\mathbf{E}(c,x) * \mathbf{F}(c)) \leq (1 - \mathbf{E}(d,x))$$
$$\bigoplus_{c \in \mathcal{N} \setminus \{d\}} (\mathbf{E}(c,x) * \mathbf{F}(c)) \oplus (\mathbf{E}(d,x) * \mathbf{F}(d)) \leq (1 - \mathbf{E}(d,x)) \oplus (\mathbf{E}(d,x) * \mathbf{F}(d))$$
$$\bigoplus_{c \in \mathcal{N}} (\mathbf{E}(c,x) * \mathbf{F}(c)) \leq (1 - \mathbf{E}(d,x)) \oplus \mathbf{F}(d)$$

which due to (1.1.11) and symmetry of $\mathbf{E}$ leads to

$$\mathbf{F}_{ANF,*}(x) \leq \mathbf{E}(x,d) \rightarrow_{\otimes} \mathbf{F}(d)$$

which together with the fact that $d \in \mathcal{N}$ has been chosen arbitrarily proves the proposition. □

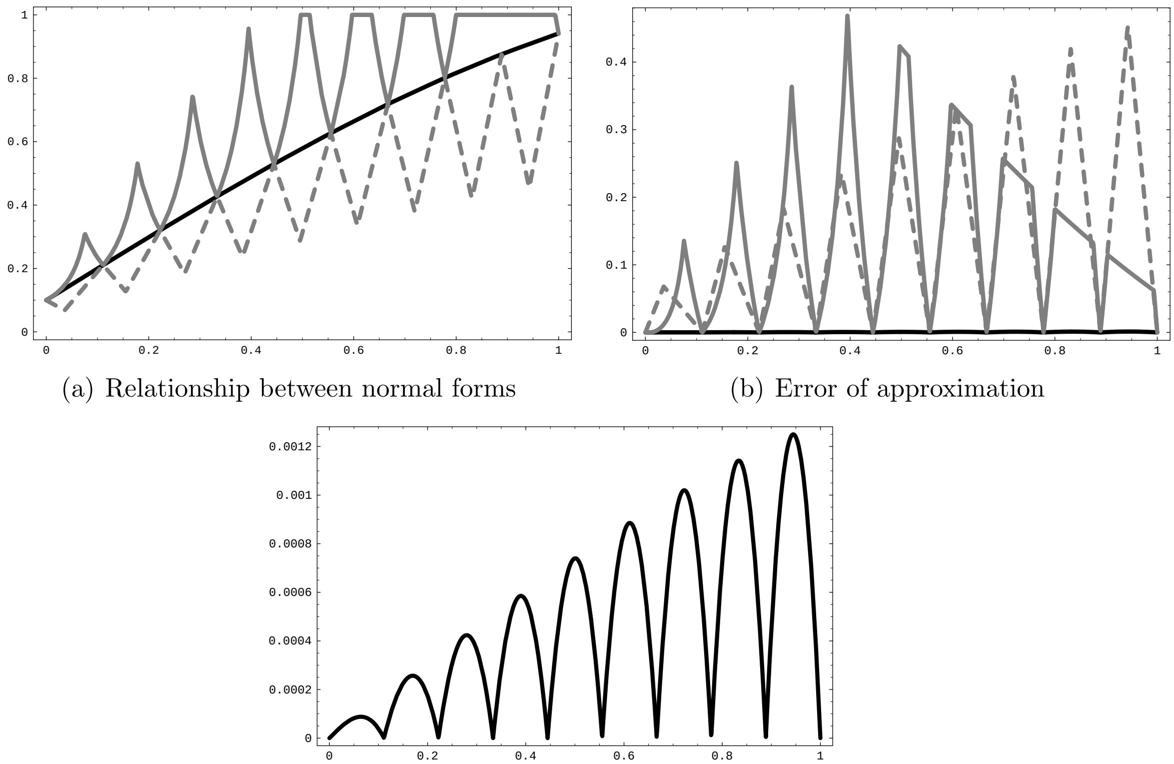
On the following example, let us illustrate the relationships between normal forms assuming a symmetrical fuzzy relation $\mathbf{E}$ as we did in Proposition 2.

Example 4 *Let us consider the fuzzy set*

$$\mathbf{F}(x) = 0.4 \sin(4x) + 0.4$$

defined on $\mathcal{D} = [0,1]$. *Furthermore, let binary fuzzy relation* $\mathbf{E}$ *be given by*

$$\mathbf{E}(x,y) = (x \leftrightarrow_{\otimes} y)^k,$$



(a) Relationship between normal forms

(b) Error of approximation

(c) Error of approximation by the additive normal form

Figure 5.2: An illustration of approximation abilities of all three given normal forms for the fuzzy set from Example 4. The black line represents $\mathbf{F}_{ANF,\odot}$, the dashed gray line represents $\mathbf{F}_{DNF,\odot}$ and the smooth gray line belongs to $\mathbf{F}_{CNF,\odot}$.

while set of nodes $\mathcal{N} = \{c_i | i = 1, \ldots, k+1\}$ contains such $c_i = (i-1)/k$. Finally, let $*$ be the product t-norm.

Then, we obtain a relationship between the conjunctive, disjunctive and additive normal forms which is illustrated on Figure 5.2 for $k = 9$. From Figure 5.2(c), it is clear that the additive normal form is absolutely the best approximation from the set of normal forms for $\mathbf{F}$ with respect to $\mathbf{E}$ and the given number and fixed distribution of nodes c_i over $\mathcal{D}$. This fact follows from Proposition 1 and Proposition 2.

In Remark 18, we have opened the question of approximation. In general, if we talk about approximation, the following three main items should be specified:

1. What we approximate, 2. By what we approximate it and 3. How we measure quality of the approximation.

In this case, we approximate fuzzy sets that are extensional with respect to a fixed fuzzy relation $\mathbf{E}$. We approximate them by fuzzy sets in normal forms which constitute a subclass of all extensional fuzzy sets. However, we have no normed or metric space, that is, we are not provided with a norm or a metric function for measuring quality of approximation. On the other hand, we are provided by a biresiduation (biresiduum) operation which expresses a measure of equivalence between two objects. Moreover, a biresiduation operation is inverse to a metric.

Thus, we do not determine upper estimations of an approximation error measured by a metric, but lower estimations of equivalence between a fuzzy set and a normal form of the given fuzzy set. Below, we prove that this lower estimations are independent on the original fuzzy set.

Theorem 7 *[14] If $\mathbf{F} \mathrel{\underset{\sim}{\subset}} \mathcal{D}$ is extensional with respect to $\mathbf{E} \mathrel{\underset{\sim}{\subset}} \mathcal{D} \times \mathcal{D}$ and $*$ then*

$$\mathbf{C}_*(x) \leq \mathbf{F}_{DNF,*}(x) \leftrightarrow_* \mathbf{F}(x), \tag{5.2.9}$$

$$\mathbf{C}_*(x) \leq \mathbf{F}_{CNF,*}(x) \leftrightarrow_* \mathbf{F}(x), \tag{5.2.10}$$

for all $x \in \mathcal{D}$, where

$$\mathbf{C}_*(x) = \bigvee_{c \in \mathcal{N}} (\mathbf{E}(x,c) * \mathbf{E}(c,x)). \tag{5.2.11}$$

PROOF: From the extensionality of $\mathbf{E}$ and the monotonicity of $*$ we obtain

$$\mathbf{E}(c,x) * \mathbf{E}(x,c) * \mathbf{F}(x) \leq \mathbf{E}(c,x) * \mathbf{F}(c),$$

therefore

$$\mathbf{C}_*(x) \leq \mathbf{F}(x) \rightarrow_* \mathbf{F}_{DNF,*}(x),$$

and together with (5.2.8), we obtain (5.2.9).

For arbitrary left continuous t-norm $*$, its residuation $\rightarrow_*$ and for $a, b, c, d \in [0, 1]$ the following

$$(a \rightarrow_* b) * (c \rightarrow_* d) \leq (a * c) \rightarrow_* (b * d)$$

holds and so, we can write

$$(\mathbf{E}(x, c) \rightarrow_* \mathbf{F}(c)) * (\mathbf{E}(c, x) \rightarrow_* \mathbf{E}(c, x)) \leq$$
$$(\mathbf{E}(x, c) * \mathbf{E}(c, x)) \rightarrow_* (F(c) * \mathbf{E}(c, x)).$$

It is clear that $x \rightarrow_* x = 1$ and $\mathbf{F}$ is extensional, thus

$$\mathbf{E}(x, c) * \mathbf{E}(c, x) \leq (\mathbf{E}(x, c) \rightarrow_* \mathbf{F}(c)) \rightarrow_* \mathbf{F}(x).$$

Finally, from (1.1.9) and using transitivity of $\rightarrow_*$, we obtain

$$\mathbf{C}_*(x) \leq \mathbf{F}_{CNF,*}(x) \rightarrow_* \mathbf{F}(x).$$

And similarly to the proof of (5.2.9), since $\mathbf{F}(x) \leq \mathbf{F}_{CNF,*}(x)$, then we obviously obtain the proof of (5.2.10). □

The following corollary (introduced in [12]) shows that in the special case, the extensionality property implies the equality of $\mathbf{F}$ to its normal form and vice-versa.

Corollary 9 *Let $\mathcal{N} = \mathcal{D}$ and fuzzy relation $\mathbf{E} \precsim \mathcal{D} \times \mathcal{D}$ be reflexive. Then $\mathbf{F} \precsim \mathcal{D}$ is extensional with respect to $\mathbf{E}$ and $*$ if and only if $\mathbf{F}_{DNF,*}(x) = \mathbf{F}(x) = \mathbf{F}_{CNF,*}(x)$ for all $x \in \mathcal{D}$.*

PROOF: By reflexivity of $\mathbf{E}$, we have $\mathbf{C}_*(x) = 1$ for each $x \in \mathcal{D}$, which proves the first implication. We obtain the reverse directly from the equality $\mathbf{F}_{DNF,*}(x) = \mathbf{F}(x) = \mathbf{F}_{CNF,*}(x)$ using properties of $\wedge$ and $\vee$. □

Theorem 7 and Proposition 2 also lead to the following obvious corollary.

Corollary 10 *Let $\mathbf{F} \subsetapprox \mathcal{D}$ be extensional with respect to symmetric fuzzy relation $\mathbf{E} \subsetapprox \mathcal{D} \times \mathcal{D}$ and $\otimes$. Moreover, let $\mathbf{E}$ fulfils the orthogonality property with respect to $\mathcal{N}$. Then*

$$\mathbf{C}_{\otimes}(x) \leq \mathbf{F}_{ANF,\otimes}(x) \leftrightarrow_{\otimes} \mathbf{F}(x), \tag{5.2.12}$$

for all $x \in \mathcal{D}$.

PROOF: Based on the fact that $\mathbf{F}_{DNF,\otimes} \leq \mathbf{F}_{ANF,\otimes} \leq \mathbf{F}_{CNF,\otimes}$, see [59]. $\square$

The question of conditional equivalence of other additive normal forms is partially clarified by the following theorem where the notion of weaker and stronger t-norm will be used.

We say that a t-norm $*_1$ is weaker than $*_2$ if $a *_1 b \leq a *_2 b$ for all $a, b \in [0,1]$. Then, we write $*_1 \leq *_2$. Analogously, we say that $*_1$ is stronger than $*_2$ if $a *_1 b \geq a *_2 b$ for all $a, b \in [0,1]$, and we write $*_1 \geq *_2$, see [38].

Theorem 8 *Let $\mathbf{E} \subsetapprox \mathcal{D} \times \mathcal{D}$ be symmetric and fulfils the orthogonality condition with respect to $\mathcal{N}$. Let $\mathbf{F} \subsetapprox \mathcal{D}$ be extensional with respect to $\mathbf{E}$ and $\otimes$, and moreover, let $\mathbf{F}$ be extensional with respect to $\mathbf{E}$ and $*$.*

(1) *If $*$ is weaker than $\otimes$ then for all $x \in \mathcal{D}$*

$$\mathbf{C}_{*}(x) \leq \mathbf{F}_{ANF,*}(x) \leftrightarrow_{*} \mathbf{F}(x). \tag{5.2.13}$$

(2) *If $*$ is stronger than $\otimes$ then for all $x \in \mathcal{D}$*

$$\mathbf{C}_{\otimes}(x) \leq \mathbf{F}_{ANF,*}(x) \leftrightarrow_{\otimes} \mathbf{F}(x). \tag{5.2.14}$$

PROOF: Using Proposition 2, Theorem 7 and properties (1.1.9) and (1.1.10) of the residuation operation we get

$$\mathbf{C}_{*}(x) \leq \mathbf{F}(x) \rightarrow_{*} \mathbf{F}_{DNF,*}(x) \leq \mathbf{F}(x) \rightarrow_{*} \mathbf{F}_{ANF,*}(x), \tag{5.2.15}$$

$$\mathbf{C}_{\otimes}(x) \leq \mathbf{F}_{CNF,\otimes}(x) \rightarrow_{\otimes} \mathbf{F}(x) \leq \mathbf{F}_{ANF,*}(x) \rightarrow_{\otimes} \mathbf{F}(x). \tag{5.2.16}$$

If $*$ is weaker than $\otimes$ then

$$\mathbf{C}_*(x) \leq \mathbf{C}_\otimes(x) \leq \mathbf{F}_{ANF,*}(x) \rightarrow_\otimes \mathbf{F}(x) \leq \mathbf{F}_{ANF,*}(x) \rightarrow_* \mathbf{F}(x),$$

which together with (5.2.15) proves (5.2.13).

If $\otimes$ is weaker than $*$ then

$$\mathbf{C}_\otimes(x) \leq \mathbf{C}_*(x) \leq \mathbf{F}(x) \rightarrow_* \mathbf{F}_{ANF,*}(x) \leq \mathbf{F}(x) \rightarrow_\otimes \mathbf{F}_{ANF,*}(x),$$

which together with (5.2.16) proves (5.2.14). □

Now, we are going to characterize the orthogonality analogously to Corollary 9, where we have characterized the extensionality.

Remark 19 *Note that if* $\mathbf{E}$ *is reflexive and* $\mathcal{N} = \mathcal{D}$ *then the orthogonality property is valid if and only if* $\mathbf{E}$ *is trivial, i.e.* $\mathbf{E}(x, y) = 1$ *for* $x = y$*, and otherwise,* $\mathbf{E}(x, y) = 0$*.*

Proposition 3 *Let* $\mathcal{N} = \mathcal{D}$*,* $\mathbf{F}(x) \in (0, 1)$ *for all* $x \in \mathcal{D}$*, and* $\mathbf{E} \mathrel{\underset{\sim}{\subset}} \mathcal{D} \times \mathcal{D}$ *be reflexive. Then,* $\mathbf{E}$ *fulfils the orthogonality condition with respect to* $\mathcal{N}$ *if and only if* $\mathbf{F}_{ANF,*}(x) = \mathbf{F}(x)$ *for all* $x \in \mathcal{D}$*.*

PROOF: Since the reflexivity and the orthogonality of $\mathbf{E}$ we obtain the following equality

$$\bigoplus_{c \in \mathcal{D} \setminus \{x\}} \mathbf{E}(c, x) = 0,$$

and therefore

$$\bigoplus_{c \in \mathcal{D}} (\mathbf{E}(c, x) * \mathbf{F}(c)) = \bigoplus_{c \in \mathcal{D} \setminus \{x\}} (\mathbf{E}(c, x) * \mathbf{F}(c)) \oplus (\mathbf{E}(x, x) * \mathbf{F}(x)) = \mathbf{F}(x).$$

On the other side, by reflexivity of $\mathbf{E}$ and $\mathbf{F}(x) \in (0, 1)$, for all $x \in \mathcal{D}$, we obtain that $\mathbf{E}(x, y) = 0$ for $x \neq y$. Thus, $\mathbf{E}$ is trivial, i.e., the orthogonality condition with respect to $\mathcal{D}$ is valid, see Remark 19. □

5.3 F-transform and the Additive Normal Form

Here, we debunk the inverse F-transform of as a special case of the additive normal form. Hence we can bring to bear all the results from the theory of the normal forms on this special additive normal form.

As the next step, we will introduce a generalized F-transform with respect to a strict t-norm for a class of continuous fuzzy sets. Taking into account the continuity of the vagueness phenomenon then the continuity of fuzzy sets is a natural requirement.

Below, we consider that $\mathcal{D} = [a, b] \subseteq \mathbb{R}$ and $*$ is a continuous strict t-norm, i.e.

$$x * y = g^{-1}(g(x) \cdot g(y)),$$

where $g : [0, 1] \to [0, 1]$ is a continuous strictly increasing mapping called *multiplicative generator*. For simplicity, we will write gx instead of $g(x)$.

Definition 36 Let $f \underset{\sim}{\subset} \mathcal{D}$ be continuous and $\mathbf{E} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}$. We say that $\mathbf{F} \underset{\sim}{\subset} \mathcal{D}$ is the *F-transform* of f with respect to $\mathbf{E}$ and $*$ if

$$\mathbf{F}(x) = g^{-1} \left(\frac{\int_{\mathcal{D}} g\mathbf{E}(x, y) \cdot gf(y)\, dy}{\int_{\mathcal{D}} g\mathbf{E}(x, y)\, dy} \right). \tag{5.3.1}$$

Remark 20 *Let x be a node c_i, $i = 1, \ldots, n$. Then since the multiplicative generator of the product operation is the identity mapping [38] we immediately come to the fact that generalized F-transform $\mathbf{F}(x)$ defined by (5.3.1) equals to the i-th component of the F-transform F_i from Definition 21. It means, that Definition 36 does not conflict the original definition of the F-transform but generalizes it from two points of view. First, it allows to deal with an arbitrary continuous strict t-norm. Second, the direct F-transform is a mapping $\mathbf{F} : \mathcal{D} \to [0, 1]$ while the original F-transform was a mapping defined on a finite set of nodes $\mathcal{N} \subset \mathcal{D}$. Thus the new definition* *is a continuous extension to the whole domain. The fact that this generalized F-transform is defined only for mappings to the unit interval is not restricting at all since any continuous mapping on a real interval can be rescaled to keep the unit interval range.*

Both following lemmas relate to the extensionality of the F-transform. The first lemma assumes that the definite integral over $\mathcal{D}$ of the similarity relation $\mathbf{E}$ mapped by generator g is a finite number i.e. there exist some maximal value $m \in \mathbb{R}$ limiting the integral.

Lemma 20 *Let* $f \underset{\sim}{\subseteq} \mathcal{D}$, $\mathbf{E}$ *be a* $*$*-similarity on* $\mathcal{D}$*, and* $\mathbf{F}$ *be the F-transform of* f *with respect to* $\mathbf{E}$*. If for all* $x \in \mathcal{D}$: $\int_{\mathcal{D}} g\mathbf{E}(x,y)\,dy = m < +\infty$ *then* $\mathbf{F}$ *is extensional with respect to* $\mathbf{E}$ *and* $*$ *on* $\mathcal{D}$*.*

PROOF: From the transitivity and symmetry of $\mathbf{E}$, we obtain

$$g\mathbf{E}(x,y) \cdot g\mathbf{E}(x,z) \leq g\mathbf{E}(y,z),$$

and obviously

$$g\mathbf{E}(x,y) \cdot \frac{1}{m} \cdot g\mathbf{E}(x,z) \cdot gf(z) \leq \frac{1}{m} \cdot g\mathbf{E}(y,z) \cdot gf(z),$$

integrating over $\mathcal{D}$ we come to

$$g\mathbf{E}(x,y)\frac{1}{m}\int_{\mathcal{D}} g\mathbf{E}(x,z) \cdot gf(z)\,dz \leq \frac{1}{m}\int_{\mathcal{D}} g\mathbf{E}(y,z) \cdot gf(z)\,dz,$$

which finally leads to

$$\mathbf{E}(x,y) \leq g^{-1}\left(\frac{\int_{\mathcal{D}} g\mathbf{E}(x,z) \cdot gf(z)\,dz}{m}\right) \rightarrow_* g^{-1}\left(\frac{\int_{\mathcal{D}} g\mathbf{E}(y,z) \cdot gf(z)\,dz}{m}\right),$$

and the reverse implication follows from symmetry of $\mathbf{E}$. □

The second lemma states the fact that the generalized F-transform is extensional with respect to the relation $\mathbf{E}^2$ even without the previous assumption.

Lemma 21 *Let $f \underset{\sim}{\subseteq} \mathcal{D}$, $\mathbf{E}$ be a $*$-similarity on $\mathcal{D}$, and $\mathbf{F}$ be the F-transform of f with respect to $\mathbf{E}$. Then $\mathbf{F}$ is extensional with respect to $\mathbf{E}^2$ and $*$ on $\mathcal{D}$.*

PROOF: From the transitivity, symmetry of $\mathbf{E}$ and the monotonicity of $\cdot$, we obtain the following inequalities

$$\int_{\mathcal{D}} g\mathbf{E}(x,y)\cdot g\mathbf{E}(x,z)\cdot gf(z)\;dz \le \int_{\mathcal{D}} g\mathbf{E}(y,z)\cdot gf(z)\;dz,$$
$$\int_{\mathcal{D}} g\mathbf{E}(x,y)\cdot g\mathbf{E}(y,z)\;dz \le \int_{\mathcal{D}} g\mathbf{E}(x,z)\;dz,$$

because we integrate over z, we have

$$g\mathbf{E}(x,y)\cdot \int_{\mathcal{D}} g\mathbf{E}(x,z)\cdot gf(z)\;dz \le \int_{\mathcal{D}} g\mathbf{E}(y,z)\cdot gf(z)\;dz,$$
$$g\mathbf{E}(x,y)\cdot \int_{\mathcal{D}} g\mathbf{E}(y,z)\;dz \le \int_{\mathcal{D}} g\mathbf{E}(x,z)\;dz,$$

and hence

$$(g\mathbf{E}(x,y))^2\cdot \frac{\int_{\mathcal{D}} g\mathbf{E}(x,z)\cdot gf(z)\;dz}{\int_{\mathcal{D}} g\mathbf{E}(x,z)\;dz} \le \frac{\int_{\mathcal{D}} g\mathbf{E}(y,z)\cdot gf(z)\;dz}{\int_{\mathcal{D}} g\mathbf{E}(y,z)\;dz},$$

by symmetry of $\mathbf{E}$, we finally conclude that $\mathbf{E}^2(x,y) \le \mathbf{F}(x) \leftrightarrow_* \mathbf{F}(y)$. □

Now, let us introduce a generalized definition of the inverse F-transform in the context of normal forms. It is based on Definition 23 where $\odot$ is used. In the following definition, we consider an arbitrary strict continuous t-norm.

Definition 37 Let $f \underset{\sim}{\subseteq} \mathcal{D}$, $\mathbf{E} \underset{\sim}{\subseteq} \mathcal{D}\times\mathcal{D}$, and $\mathcal{N} \subseteq \mathcal{D}$. Furthermore, let $\mathbf{F}$ be the F-transform of f with respect to $\mathbf{E}$ and $*$. Then the additive normal form of $\mathbf{F}$ with respect to $\mathbf{E}$ and $*$

$$f_{FT}(x) = \bigoplus_{c\in\mathcal{N}} (\mathbf{E}(c,x) * \mathbf{F}(c)) \tag{5.3.2}$$

will be called the *inverse F-transform* of f with respect to $\mathbf{E}$ and $*$.

So, we defined a generalized direct F-transform which is moreover an extension of the original one to the whole domain. We further applied the additive normal form to this extended mapping to get the inverse F-transform. Again, there is no conflict with the original definition.

In the following proposition, we show the conditional equivalence for the F-transform which is an immediately inherited result from the field of normal forms which justifies our efforts to connect both main streams belonging to the topics studied by the fuzzy approximation.

Proposition 4 *Let f, $\mathbf{F}$, $*$, $\mathcal{N}$ be as above. Let $\mathbf{E}$ be symmetric and satisfies the orthogonality condition with respect to $\mathcal{N}$ and moreover $\int_{\mathcal{D}} g\mathbf{E}(x,y)\,dy = m < +\infty$ for arbitrary $x \in \mathcal{D}$.*

(1) *If $*$ is stronger than $\otimes$ then for all $x \in \mathcal{D}$*

$$\mathbf{C}_{\otimes}(x) \le \mathbf{F}(x) \leftrightarrow_{\otimes} f_{FT}(x). \tag{5.3.3}$$

(2) *If $*$ is weaker than $\otimes$ and $\mathbf{F}$ is extensional with respect to $\mathbf{E}$ and $\otimes$ then for all $x \in \mathcal{D}$*

$$\mathbf{C}_{*}(x) \le \mathbf{F}(x) \leftrightarrow_{*} f_{FT}(x). \tag{5.3.4}$$

PROOF: From Lemma 20, we know that $\mathbf{F}$ is extensional with respect to $\mathbf{E}$ and $*$. If $*$ is stronger than $\otimes$ then $\mathbf{F}$ is extensional with respect to $\mathbf{E}$ and $\otimes$. Hence, applying Theorem 8, we obtain (5.3.3). □

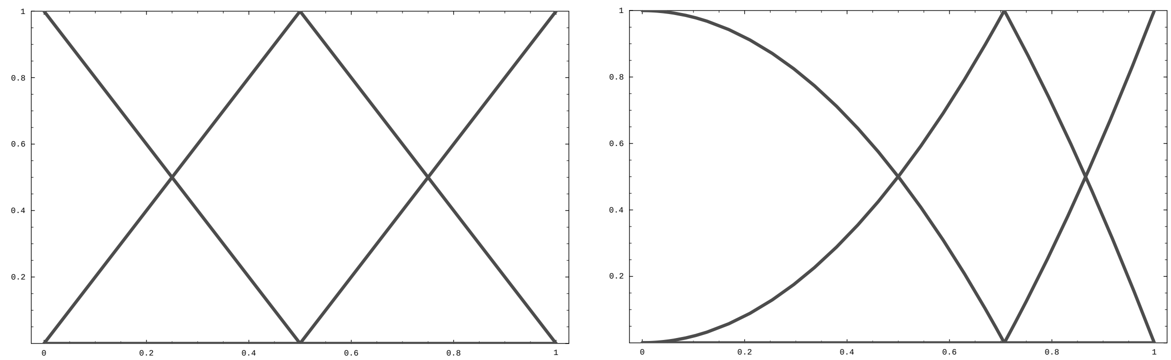
Note that $\odot$ is stronger than $\otimes$. Therefore, the quality estimation (5.3.3) is valid even for the "classical" F-transform.

Let $f \underset{\sim}{\subseteq} \mathcal{D}$ and $\mathbf{E}_k \underset{\sim}{\subseteq} \mathcal{D} \times \mathcal{D}$ be defined as follows

$$\mathbf{E}_k(x,y) = (T(x) \leftrightarrow_{*} T(y))^k, \tag{5.3.5}$$

where $k \in \mathbb{N}$ and $T : \mathcal{D} \to [0,1]$ is given by

$$T(x) = \frac{x-a}{b-a}. \tag{5.3.6}$$



(a) $\mathcal{D} = [0, 1]$ and $T(x)$ is given by (5.3.6)

(b) $\mathcal{D} = [0, 1]$ and $T(x) = x^2$

Figure 5.3: An example of basic functions given as $\mathbf{E}_k(c_i, x)$, where c_i for $i = 1, \ldots, k+1$ are distributed over $[a, b]$ such that $\mathbf{E}_k$ fulfills the orthogonality property.

It is obvious that $\mathbf{E}_k$ is a similarity relation for each $k \in \mathbb{N}$. In general, T can be an arbitrary continuous strictly increasing function such that $T(a) = 0$ and $T(b) = 1$.

In the case of $\mathbf{E}_k$ given by (5.3.5) and $* = \otimes$, the orthogonality leads to $(k+1)$ equidistant nodes $\hat{c}_i = (i-1)/k$, $i = 1, \ldots, (k+1)$ on $[0, 1]$, which define nodes $c_i \in \mathcal{D}$ as $c_i = T^{-1}(\hat{c}_i)$. Let us denote $\mathcal{N}_k = \{c_i |\ i = 1, \ldots, k+1\}$.

It is worth mentioning that fuzzy relations $\mathbf{E}_k(c_i, x)$, where nodes c_i are chosen to fulfil the orthogonality condition, determine the basic functions from Definition 19 of the triangular shape whenever $\otimes$ is considered (see Fig. 5.3). Moreover, let us stress that values $\mathbf{F}_k(c_i)$ where $\mathbf{F}_k$ is the F-transform with respect to $\mathbf{E}_k$ are exactly equal to the components F_i of the F-transform from Definition 21.

Let $\mathbf{F}_k$ denote the F-transform with respect to $\mathbf{E}_k$ for an arbitrary $k \in \mathbb{N}$, and moreover

$$\mathbf{E}_\infty(x, y) = \lim_{k\to\infty} \mathbf{E}_k(x, y) = \begin{cases} 1 & x = y, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbf{F}_\infty(x) = \lim_{k\to\infty} \mathbf{F}_k(x) = f(x),$$

If $\mathcal{N} = \mathcal{D}$ then we obtain immediately that

$$\mathbf{F}_{ANF,*}(x) = \bigoplus_{c \in \mathcal{D}} (\mathbf{E}_\infty(c, x) * \mathbf{F}_\infty(c)) = f(x). \tag{5.3.7}$$

All results from this section have been established for the one-dimensional case with the aim of better transparency. Nevertheless, a generalization is straightforward and leads to the following formula

$$f_{FT}(\mathbf{x}) = \bigoplus_{\mathbf{c} \in \mathcal{N}} (\mathbf{E}(\mathbf{c}, \mathbf{x}) * \mathbf{F}(\mathbf{c})),$$

where $\mathbf{F}$ is given by

$$\mathbf{F}(\mathbf{x}) = g^{-1}\left(\frac{\int_{\mathcal{D}} g\mathbf{E}(\mathbf{x}, \mathbf{y}) * gf(\mathbf{y})\, d\mathbf{y}}{\int_{\mathcal{D}} g\mathbf{E}(\mathbf{x}, \mathbf{y})\, d\mathbf{y}} \right)$$

fuzzy relation $\mathbf{E}$ is given as follows $\mathbf{E}(\mathbf{x}, \mathbf{y}) = \mathbf{E}(x_1, y_1) * \ldots * \mathbf{E}(x_n, y_n)$ and the domain $\mathcal{D}$ is an n-dimensional cube.

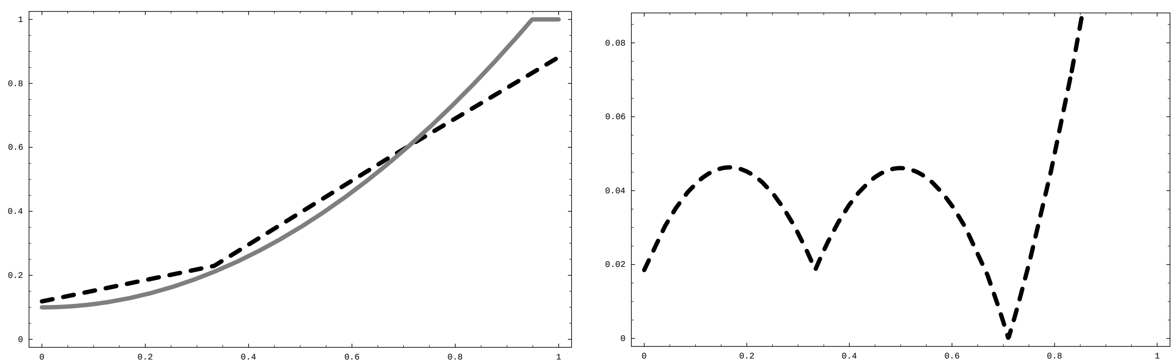
Let us illustrate properties of the F-transform f_{FT} on the following example.

Example 5 *Let $f(x) = 1 \wedge (x^2 + 0.1)$ on $\mathcal{D} = [0, 1]$ and let $*$ be the product t-norm $\odot$.*

Considering $k = 3$, we obtain that

$$|\mathbf{F}_k(x) - f_{FT}(x)| \leq 1 - \bigvee_{i=1}^{4} \mathbf{E}_3^2(c_i, x).$$

The final approximation is depicted on Fig. 5.4.



(a) Original function and its approximation for $k = 3$

(b) Error of approximation given by $e(x) = |f_{FT}(x) - f(x)|$

Figure 5.4: Example of f_{FT} given by (5.3.2) for $f(x) = 1 \wedge (x^2 + 0.1)$ on $\mathcal{D} = [0, 1]$.

Chapter 6

Additive Interpretation of an FRB

Let us again recall that unless stated otherwise, we will assume $*$ to be a left-continuous t-norm and $\rightarrow_*$ its adjoint residuation and $\oplus$ stands for the Łukasiewicz t-conorm.

An expert knowledge is often expressed in the form of an FRB which is comprised from n fuzzy rules of type (1.2.2) or of type (1.2.4) where $\mathcal{A}_i$ and $\mathcal{F}_i$ are linguistic expressions represented by fuzzy sets $\mathbf{A}_i \underset{\sim}{\subset} \mathcal{D}$ and $\mathbf{F}_i \underset{\sim}{\subset} \mathcal{D}'$, respectively.

The FRB consisting of n fuzzy rules (1.2.2) is usually interpreted by the fuzzy relation $\hat{\mathbf{R}}_* \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ given by

$$\hat{\mathbf{R}}_*(x, y) = \bigwedge_{i=1}^{n} (\mathbf{A}_i(x) \rightarrow_* \mathbf{F}_i(y))$$

and the FRB consisting of n fuzzy rules (1.2.4) is usually interpreted by the fuzzy relation $\check{\mathbf{R}}_* \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ given by

$$\check{\mathbf{R}}_*(x, y) = \bigvee_{i=1}^{n} (\mathbf{A}_i(x) * \mathbf{F}_i(y)).$$

As mentioned in the previous chapter, interpretations of FRB's relate to normal forms, fuzzy relation $\hat{\mathbf{R}}_*$ to the conjunctive normal form and $\check{\mathbf{R}}_*$ to the disjunctive normal form, in particular. The development of the additive normal forms introduced in [59] and continued in [14] (see Chapter 5) directly leads to an investigation of new possible FRB interpretations related to these normal forms. Furthermore, such investigation connects the F-transform even to the field of FRB interpretations and related applications such as e.g. fuzzy control since the F-transform has been proven to be a special case of the additive normal form.

Let us recall that T. Takagi and M. Sugeno [76] have proposed fuzzy rules in the form given by (1.2.10) which are interpreted by weighted arithmetic mean (1.2.11) of the consequent parts of the rules.

If we consider Takagi-Sugeno rules with constant consequents $F_i \in \mathbb{R}$ and the antecedent fuzzy sets $\mathbf{A}_i \underset{\sim}{\subseteq} \mathcal{D}$ fulfilling the *Ruspini* condition [62]:

$$\sum_{i=1}^{n} \mathbf{A}_i(x) = 1, \quad \text{for } x \in \mathcal{D} \tag{6.0.1}$$

the rules will be interpreted by the following formula

$$\sum_{i=1}^{n} \mathbf{A}_i(x) F_i. \tag{6.0.2}$$

This shows why functions produced by the the F-transform can be treated as Takagi-Sugeno models of the 0-th order.

Let us consider crisp values F_i to be singletons i.e. special fuzzy numbers on $\mathcal{D}'$. Since the product is a particular t-norm and we assume the Ruspini condition we can introduce the following definition where singletons F_i are replaced by fuzzy sets in general and the product t-norm is replaced by an arbitrary one.

Definition 38 [70] Let us be given FRB (1.2.4) and let expressions $\mathcal{A}_i, \mathcal{F}_i$ are represented by fuzzy sets $\mathbf{A}_i \underset{\sim}{\subseteq} \mathcal{D}$ and $\mathbf{F}_i \underset{\sim}{\subseteq} \mathcal{D}'$ for $i = 1, ..., n$, respectively. Then the following fuzzy relation

$$\mathbf{R}_*^{\oplus}(x, y) = \bigoplus_{i=1}^{n} (\mathbf{A}_i(x) * \mathbf{F}_i(y)) \tag{6.0.3}$$

will be called the *additive interpretation* of the given FRB.

Fuzzy relation $\mathbf{R}_*^{\oplus}$ stems from Takagi-Sugeno models but it is also closely related to the fuzzy relation $\check{\mathbf{R}}_*$ interpreting FRB (1.2.4) where the difference is given only by another t-conorm aggregating all rules. Hence, $\mathbf{R}_*^{\oplus}$ given by (6.0.3) is an interpretation of FRB (1.2.4) [67]. Moreover, it has a direct connection to a variety of relations appearing in neuro-fuzzy systems [26].

The foregoing sections are devoted to the study of the additive interpretations, particularly to approximation and interpolation properties and to relationship between the additive interpretations and the F-transform.

6.1 Approximation Properties

The universal approximation property is a typical request to FRB systems [6, 7, 40, 52] and it is a well-known fact that the two standard interpretations of FRBs $\hat{\mathbf{R}}_*$ and $\check{\mathbf{R}}_*$ *ε-approximate* an arbitrary continuous function on a compact domain. Here we claim that the additive interpretation of an FRB keeps the same property. First, let us recall the following definition.

Definition 39 [52] Let $\mathbf{R}$ be a fuzzy relation on $\mathcal{D}\times\mathcal{D}'$, $f : \mathcal{D} \to \mathcal{D}'$ be a continuous function and $\varepsilon > 0$. We say that the relation $\mathbf{R}$ ε-approximates the function f if

$$\forall x \in \mathcal{D}, \forall y \in \mathcal{D}' : \quad \mathbf{R}(x, y) > 0 \Rightarrow \mid y - f(x) \mid < \varepsilon.$$

And the universal approximation property for the additive interpretations can be formulated as follows.

Theorem 9 *[70] Let $\mathcal{D}, \mathcal{D}'$ be two closed real intervals and let $f : \mathcal{D} \to \mathcal{D}'$ be an arbitrary continuous function. Then for arbitrary $\varepsilon > 0$ there exist $\mathbf{A}_i \underset{\sim}{\subseteq} \mathcal{D}$, $\mathbf{F}_i \underset{\sim}{\subseteq} \mathcal{D}'$ and $n \in \mathbb{N}$ such that the fuzzy relation given by (6.0.3) ε-approximates f.*

PROOF: The technique of the proof stems from the proof of universal approximation property for two standard interpretations published in [52].

Let $\mathcal{D} = [a, b]$. The continuity of f on $\mathcal{D}$ means that f is uniformly continuous:

$$\forall \varepsilon\ \exists \delta:\ \forall x, x' \in \mathcal{D}\ \mid x - x' \mid < \delta \Rightarrow \mid f(x) - f(x') \mid < \varepsilon/2. \qquad (6.1.1)$$

Let us fix an arbitrary $\varepsilon > 0$ and find $\delta > 0$ with respect to 6.1.1. Let us choose $n > 2$ such that

$$h = \frac{\mid b - a \mid}{n - 1} < \frac{\delta}{2}.$$

Denote $x_1 = a$ and $x_i = x_{i-1} + h$ for $i = 2, \ldots, n$.

Let $\mathbf{A}_i \underset{\sim}{\subset} \mathcal{D}$ for $i = 1, \ldots, n$ be such that

- $\mathbf{A}_i(x_i) = 1$ for
- $\mathbf{A}_i(x) > 0$ if and only if $x \in (x_{i-1}, x_{i+1})$ where $x_0 = x_1, x_{n+1} = x_n$.

Let $U_i = (x_{i-1}, x_{i+1})$ then $f(U_i) = [a_i, b_i]$ where $a_i \leq b_i$. Denote the center of the interval $[a_i, b_i]$ by y_i and let us construct interval $V_i \subseteq \mathcal{D}'$ as follows

$$V_i = (y_i - \varepsilon/2, y_i + \varepsilon/2).$$

Because $|x_{i-1} - x_{i+1}| < \delta$ and because of the continuity of f we get $\mid b_i - a_i \mid < \varepsilon/2$ and therefore $f(U_i) \subset V_i$.

Let $\mathbf{F}_i \underset{\sim}{\subset} \mathcal{D}'$ be such that $\mathbf{F}_i(y) > 0$ if and only if $y \in V_i$ and take an arbitrary $x' \in \mathcal{D}$ such that $x' \notin \{x_i \mid i = 1, \ldots, n\}$. Then there exist U_i, U_{i+1} such that $x' \in U_i$ and $x' \in U_{i+1}$.

If $x' \in U_i$ and $x' \in U_{i+1}$ then $\mathbf{A}_i(x') > 0$ as well as $\mathbf{A}_{i+1}(x') > 0$ while $\mathbf{A}_j(x') = 0$ for $j \notin \{i, i+1\}$ and furthermore, $f(x') \in V_i$ and $f(x') \in V_{i+1}$.

Take an arbitrary $y \in \mathcal{D}'$ such that $\mathbf{R}_*^{\oplus}(x', y) > 0$. Then

$$((\mathbf{A}_i(x') * \mathbf{F}_i(y)) \oplus (\mathbf{A}_{i+1}(x') * \mathbf{F}_{i+1}(y))) > 0$$

which occurs if and only if $\mathbf{F}_i(y) > 0$ or $\mathbf{F}_{i+1}(y) > 0$. Without loss of generality let $\mathbf{F}_i(y) > 0$ then $y \in V_i$. Finally, we can write

$$| y - f(x') |=| y - y_i + y_i - f(x') |\leq| y - y_i | + | y_i - f(x') |\leq \varepsilon.$$

In case of $x' \in \{x_i \mid i = 1, \ldots, n\}$ the proof uses the same technique and therefore it is omitted. $\square$

Some results and techniques from Chapter 5 can be very helpful. For instance, the orthogonality condition which played a crucial role in Chapter 5 can be slightly modified to allow us to use it in the context of an FRB interpretations.

Definition 40 We say that $\mathbf{A}_i \underset{\sim}{\subset} \mathcal{D},\ i = 1, \ldots, n$ fulfill the orthogonality condition if

$$\bigoplus_{\substack{i=1\\i\neq j}}^{n} \mathbf{A}_i(x) = 1 - \mathbf{A}_j(x), \quad j = 1, \ldots, n. \tag{6.1.2}$$

The Ruspini condition seems to be essential for the additive interpretations given by (6.0.3). The following lemma characterizes the orthogonality condition as equivalent to the Ruspini condition.

Lemma 22 *[67] Fuzzy sets $\mathbf{A}_i \in \mathcal{F}(\mathcal{D}),\ i = 1, \ldots, n$ fulfill the orthogonality condition if and only if*

$$\sum_{i=1}^{n} \mathbf{A}_i(x) = 1, \quad x \in \mathcal{D}. \tag{6.1.3}$$

PROOF: Let $j \in \{1, \ldots, n\}$ be an arbitrary index.

First, let $x \in \mathcal{D}$ be such that $\mathbf{A}_j(x) \in (0, 1)$. Then

$$\bigoplus_{\substack{i=1\\i\neq j}}^{n} \mathbf{A}_i(x) = \sum_{\substack{i=1\\i\neq j}}^{n} \mathbf{A}_i(x)$$

and therefore (6.1.3) is fulfilled.

Second, let $x \in \mathcal{D}$ be such that $\mathbf{A}_j(x) = 1$. Then from the orthogonality condition, $\mathbf{A}_i(x) = 0$ for any index $i \neq j$ and therefore condition (6.1.3) is fulfilled.

Third, let $x \in \mathcal{D}$ be such that $\mathbf{A}_j(x) = 0$. Then from the orthogonality condition, there necessarily exists another index $k \neq j$ such that either $\mathbf{A}_k(x) \in (0, 1)$ or $\mathbf{A}_k(x) = 1$. The rest goes as above.

The other side of the proof, showing that condition (6.1.3) implies the orthogonality, is trivial and therefore it is omitted. □

Just as we specified mutual position of normal forms by Proposition 2, we can clarify a mutual position between the additive interpretation and standard interpretations of an FRB by the following proposition.

Proposition 5 *[70] Let us be given pairs of input-output fuzzy sets* $(\mathbf{A}_1, \mathbf{F}_1), \ldots, (\mathbf{A}_n, \mathbf{F}_n)$ *where* $\mathbf{A}_i \underset{\sim}{\subset} \mathcal{D}$ *and* $\mathbf{F}_i \underset{\sim}{\subset} \mathcal{D}'$. *Let* $\mathbf{A}_i \underset{\sim}{\subset} \mathcal{D}$ *fulfill the orthogonality condition given by (6.1.2). Then the following inequalities hold*

$$\bigvee_{i=1}^{n}(\mathbf{A}_i(x) * \mathbf{F}_i(y)) \leq \bigoplus_{i=1}^{n}(\mathbf{A}_i(x) * \mathbf{F}_i(y)) \leq \bigwedge_{i=1}^{n}(\mathbf{A}_i(x) \rightarrow_{\otimes} \mathbf{F}_i(y)). \tag{6.1.4}$$

PROOF: The proof uses the same technique as the proof of Proposition 2 and therefore it is omitted. □

6.2 Interpolation Properties

6.2.1 Systems of Fuzzy Relation Equations

In Subsection 1.2.2, fuzzy interpolation and its relationship to systems of relation equations (FRE for short) has been remarked. It has been noted that the output $\mathbf{B} \underset{\sim}{\subset} \mathcal{D}'$ of a fuzzy inference mechanism is given by an image of the input $\mathbf{A} \underset{\sim}{\subset} \mathcal{D}$ under a fuzzy relation $\mathbf{R} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ which interprets a given FRB.

In most practical cases when modelling an FRB system, we deal with the direct image (sup-$*$ composition) which stems from the *compositional rule of inference* introduced by L.A. Zadeh [79]. It is worth mentioning that its logical background coincides with the *generalized modus ponens* [32].

A fuzzy rule base may be viewed as a partial mapping from $\mathcal{F}(\mathcal{D})$ to $\mathcal{F}(\mathcal{D}')$. Building a fuzzy inference module on the base of a rule base means extending this partial function to a total function. It means that in some "reasonable manner", we have to associate with an arbitrary $\mathbf{A} \underset{\sim}{\subseteq} \mathcal{D}$ some $\mathbf{B} \underset{\sim}{\subseteq} \mathcal{D}'$. The "reasonable manner" means extending the partial mapping to a total one in such a way which based on an input $\mathbf{A}_i \underset{\sim}{\subseteq} \mathcal{D}$ would determine an output precisely equal to $\mathbf{F}_i \underset{\sim}{\subseteq} \mathcal{D}'$ for $i = 1, \ldots, n$.

It leads to the following *system of FREs with the sup-$*$ composition*

$$\mathbf{A}_i \circ_* \mathbf{R} = \mathbf{F}_i \quad i = 1, \ldots, n \tag{6.2.1}$$

where $\mathbf{A}_i \underset{\sim}{\subseteq} \mathcal{D}, \mathbf{F}_i \underset{\sim}{\subseteq} \mathcal{D}'$ and $\mathbf{R} \underset{\sim}{\subseteq} \mathcal{D} \times \mathcal{D}'$.

Fuzzy relation $\mathbf{R} \underset{\sim}{\subseteq} \mathcal{D} \times \mathcal{D}'$ which fulfills the equality (6.2.1) is a *solution* to the system of FREs with the sup-$*$ composition.

Let us recall some basic results concerning the system of FREs with the sup-$*$ composition which can be found e.g. in [15, 35, 39].

Theorem 10 *System (6.2.1) is solvable if and only if $\hat{\mathbf{R}}_*$ is a solution to the system and moreover, $\hat{\mathbf{R}}_*$ is the greatest solution to (6.2.1).*

Theorem 10 is a crucial theorem in the field of system of FREs with the sup-$*$ composition. Besides the fact, that it is necessary and a sufficient condition for the solvability of system (6.2.1), it already determines the solution and moreover, it ensures the given solution to be the greatest solution. Finally, its further importance is as follows. Whenever we deal with fuzzy rules (1.2.2) interpreted by $\hat{\mathbf{R}}_*$, the direct image is the first choice for an inference mechanism since fuzzy relation $\hat{\mathbf{R}}_*$ has a unique position in the set of all possible solutions to the corresponding system of FREs with the sup-$*$ composition.

Let us recall a theorem published in [16] and then independently re-found in [35] specifying conditions upon which even $\check{\mathbf{R}}_*$ can be a solution to system (6.2.1). Moreover, it is a sufficient condition for the solvability of the system.

Theorem 11 *Let $\mathbf{A}_i \underset{\sim}{\subset} \mathcal{D}$ for $i = 1, \ldots, n$ be normal. Then $\check{\mathbf{R}}_*$ is a solution to (6.2.1) if and only if the following condition*

$$\bigvee_{x \in \mathcal{D}} (\mathbf{A}_i(x) * \mathbf{A}_j(x)) \leq \bigwedge_{y \in \mathcal{D}'} (\mathbf{F}_i(y) \leftrightarrow_* \mathbf{F}_j(y)) \tag{6.2.2}$$

holds for arbitrary $i, j \in \{1, \ldots, n\}$.

Theorem 11 specifies a condition under which, $\check{\mathbf{R}}_*$ connected to the sup-$*$ inference, is an appropriate interpretation of an FRB.

Another inference mechanism that can be treated is related to the subdirect image (inf$\rightarrow_*$ composition). The subdirect image, conversely to the direct image, does not have such a connection to the generalized modus ponens deductive rule and its motivation was different, see [3]. On the other hand, as mentioned in [31], the inference mechanism does not have to be necessarily logical but just a mapping from $\mathcal{F}(\mathcal{D})$ to $\mathcal{F}(\mathcal{D}')$ fulfilling the required properties.

The *systems of FREs with inf$\rightarrow_*$ composition* is given as follows

$$\mathbf{A}_i \triangleleft_* \mathbf{R} = \mathbf{B}_i \quad i = 1, \ldots, n \tag{6.2.3}$$

where $\mathbf{A}_i \underset{\sim}{\subset} \mathcal{D}, \mathbf{F}_i \underset{\sim}{\subset} \mathcal{D}'$ and $\mathbf{R} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$.

In the sequel, we will recall some basic facts [15, 39] about system (6.2.3) which should justify our further usage of the subdirect image as an inference mechanism.

Theorem 12 *System (6.2.3) is solvable if and only if* $\check{\mathbf{R}}_*$ *is a solution to the system and moreover,* $\check{\mathbf{R}}_*$ *is the least solution to system (6.2.3).*

Furthermore, both systems of fuzzy relation equations are dual [15]. Only for the completeness of the basic facts, let us recall the following theorem [45] which is dual to Theorem 11.

Theorem 13 *Let* $\mathbf{A}_i$ *for* $i = 1, \ldots, n$ *be normal. Then* $\hat{\mathbf{R}}_*$ *is a solution to (6.2.3) if and only if the following condition*

$$\bigvee_{x \in \mathcal{D}} (\mathbf{A}_i(x) * \mathbf{A}_j(x)) \leq \bigwedge_{y \in \mathcal{D}} (\mathbf{F}_i(y) \leftrightarrow_* \mathbf{F}_j(y))$$

holds for arbitrary $i, j \in \{1, \ldots, n\}$.

Here we observe that $\check{\mathbf{R}}_*$ has precisely the same position among other solutions to system (6.2.3) as fuzzy relation $\hat{\mathbf{R}}_*$ had in the case of system (6.2.1). So, if we adopt the idea of [31] that the inference mechanism can be understood only as certain mapping between collections of fuzzy sets without deeper connection to a logical deduction rule, nothing prevents us from treating the subdirect image as a mathematical basis for an inference mechanism.

Based on the facts and theorems above, we claim that fuzzy relation $\check{\mathbf{R}}_*$ should be primarily treated together with the subdirect image and fuzzy relation $\hat{\mathbf{R}}_*$ should be primarily treated together with the direct image. Consequently, we claim that fuzzy rules (1.2.2) are predetermined to be used with the inference based on the direct image while fuzzy rules (1.2.4) are predetermined to be used with the inference based on the subdirect image.

Remark 21 *It is worth mentioning that checking condition (6.2.2) appearing in Theorem 11 and Theorem 13 is not very convenient from a practical point of view.*

On the other hand, if the antecedent fuzzy sets form so-called $*$ semi-partition [17], fulfilment of the discussed condition is ensured in advance.

Much work has been done in the field of fuzzy relation equations [15, 30, 35], mainly aiming at the solvability criteria, greatest and lower solutions, least and greater solutions, respectively and at determining complete solution sets.

Unfortunately, this work has not been followed by practitioners very often. This was perhaps, besides other unspecified reasons, because neuro-fuzzy systems [26] and Takagi-Sugeno rules [76] are the most frequently used fuzzy methods currently. But only two particular solutions $\hat{\mathbf{R}}_*$ and $\check{\mathbf{R}}_*$ have been primarily studied [55], so far.

Therefore, an investigation of the additive interpretation of a fuzzy rule base in the context of fuzzy interpolation seems to be straightforward.

6.2.2 Systems of FRE with inf$\to_*$ composition

Fuzzy relation $\mathbf{R}_*^{\oplus}$ corresponds to fuzzy rules (1.2.4) which are assumed to be primarily related to the inference mechanism based on the subdirect image. This subsection investigates this relationship through FREs with the inf$\to_*$ composition.

Theorem 14 *[67, 68] Let $\mathbf{A}_i$, $i = 1, \ldots, n$ be normal and fulfill Ruspini condition (6.1.3). Then system (6.2.3) is solvable and $\mathbf{R}_*^{\oplus}$ is a solution.*

PROOF:

Let $j \in \{1, \ldots, n\}$ be an arbitrary fixed subindex and let

$$\mathbf{B}(y) = \bigwedge_{x \in \mathcal{D}} \left(\mathbf{A}_j(x) \to_* \bigoplus_{i=1}^{n} (\mathbf{A}_i(x) * \mathbf{F}_i(y)) \right).$$

Then

$$\mathbf{B}(y) \geq \bigwedge_{x\in X}\left(\mathbf{A}_j(x) \to_* \bigvee_{i=1}^{n}(\mathbf{A}_i(x) * \mathbf{F}_i(y))\right) \geq \bigwedge_{x\in\mathcal{D}} (\mathbf{A}_j(x) \to_* (\mathbf{A}_j(x) * \mathbf{F}_j(y))) \geq \mathbf{F}_j(y)$$

which yields $\mathbf{B} \supseteq \mathbf{F}_j$.

On the other hand

$$\mathbf{B}(y) \leq \bigwedge_{x\in\mathcal{D}}\left(\mathbf{A}_j(x) \to_* \left(\bigoplus_{\substack{i=1\\ i\neq j}}^{n}(\mathbf{A}_i(x)) \oplus \mathbf{F}_j(y)\right)\right)$$

holds. The Ruspini condition due to Lemma 22 yields

$$\mathbf{B}(y) \leq \bigwedge_{x\in\mathcal{D}}(\mathbf{A}_j(x) \to_* ((1-\mathbf{A}_j(x)) \oplus \mathbf{F}_j(y))).$$

Let $x' \in \mathcal{D}$ be such that $\mathbf{A}_j(x') = 1$ then

$$\mathbf{B}(y) \leq (\mathbf{A}_j(x') \to_* ((1-\mathbf{A}_j(x')) \oplus \mathbf{F}_j(y))) = \mathbf{F}_j(y)$$

which yields $\mathbf{B} \subseteq \mathbf{F}_j$. □

Due to Theorem 12 we can state the following corollary of Theorem 14.

Corollary 11 *[67, 68] Let $\mathbf{A}_i$, $i = 1,\dots,n$ be normal and fulfill Ruspini condition (6.1.3). Then $\check{\mathbf{R}}_*$ is a solution to system (6.2.3).*

Below, Proposition 6 claims that $\mathbf{R}_*^{\oplus}$ is not the only additive interpretation which is a solution to system (6.2.3).

Proposition 6 *[67, 68] Let $\mathbf{A}_i$, $i = 1,\dots,n$ be normal and fulfill Ruspini condition (6.1.3). Furthermore, let $\blacktriangle$ be a t-norm such that $* \leq \blacktriangle$. Then the fuzzy relation $\mathbf{R}_{\blacktriangle}^{\oplus}$ is a solution to (6.2.3).*

PROOF: The proof uses the same technique as the proof of Theorem 14 and is therefore omitted. □

Let us briefly summarize results from this subsection. Theorem 14 provides us with an easy-to-check condition certifying a proper performance of an additive interpretation of an FRB connected to a subdirect image inference mechanism. Moreover, the assumption refers only to the antecedent fuzzy sets so, its fulfillment can be ensured in advance during an identification process.

Since no solvability is assumed, the theorem impact is even in specifying sufficient solvability condition. This consequently means that $\check{\mathbf{R}}_*$ is a solution as well. Finally, a wide variety of t-norms can be used in the additive interpretations.

6.2.3 Systems of FREs with sup-$*$ composition

This subsection focuses on systems of FREs with sup-$*$ composition. The disjunctive interpretation $\check{\mathbf{R}}_*$ has been proven to be a solution to system (6.2.1) assuming certain conditions. Similarly, we study the additive interpretations on the same system.

Theorem 15 *[67, 68] Let $\mathbf{A}_i$ for $i = 1, \dots, n$ be normal and fulfill Ruspini condition (6.1.3). Furthermore, let $* \leq \otimes$ where $\otimes$ is the Łukasiewicz t-norm. Then system (6.2.1) is solvable and $\mathbf{R}_*^{\oplus}$ is a solution.*

PROOF: Let $j \in \{1, \dots, n\}$ be an arbitrary fixed subindex and let

$$\mathbf{B}(y) = \bigvee_{x \in \mathcal{D}} \left(\mathbf{A}_j(x) * \bigoplus_{i=1}^{n} (\mathbf{A}_i(x) * \mathbf{F}_i(y)) \right).$$

Let $x \in \mathcal{D}$ be such that $\mathbf{A}_j(x') = 1$. Then by direct assignments one gets

$$\mathbf{B}(y) \geq \bigvee_{x \in \mathcal{D}} \left(\mathbf{A}_j(x) * \bigvee_{i=1}^{n} (\mathbf{A}_i(x) * \mathbf{F}_i(y)) \right) \geq (\mathbf{A}_j(x') * (\mathbf{A}_j(x') * \mathbf{F}_j(y)))$$

which yields $\mathbf{B} \supseteq \mathbf{F}_j$.

On the other hand,

$$\mathbf{B}(y) \leq \bigvee_{x \in \mathcal{D}} \left(\mathbf{A}_j(x) * \left(\bigoplus_{\substack{i=1 \\ i \neq j}}^{n} (\mathbf{A}_i(x)) \oplus \mathbf{F}_j(y) \right) \right)$$

holds and the Ruspini condition yields

$$\mathbf{B}(y) \leq \bigvee_{x\in\mathcal{D}} (\mathbf{A}_j(x) * ((1 - \mathbf{A}_j(x)) \oplus \mathbf{F}_j(y)))\,.$$

Since $(1-a) \oplus b = a \rightarrow_{\otimes} b$ and since $* \leq \otimes$ we obtain

$$\mathbf{B}(y) \leq \bigvee_{x\in\mathcal{D}} (\mathbf{A}_j(x) \otimes (\mathbf{A}_j(x) \rightarrow_{\otimes} \mathbf{F}_j(y)))$$

and finally, because $a * (a \rightarrow_* b) \leq b$ for any $*$, we get $\mathbf{B} \subseteq \mathbf{F}_j$. □

Due to Theorem 10 we can state the following corollary of Theorem 15.

Corollary 12 *[67, 68] Let $\mathbf{A}_i$ for $i = 1, \ldots, n$ be normal and fulfill Ruspini condition (6.1.3). Furthermore, let $* \leq \otimes$. Then $\hat{\mathbf{R}}_*$ is a solution to system (6.2.1).*

Theorem 15 requires the use of a t-norm which is even weaker than the Łukasiewicz one, which is already a very weak t-norm, see [38]. Thus for practical applications, perhaps only the case when $* = \otimes$ is worth mentioning. In this case, the Łukasiewicz t-norm is used for both, the sup-$\otimes$ composition as an inference method and for connecting antecedent and consequent fuzzy sets in the corresponding fuzzy relation $\mathbf{R}^{\oplus}_{\otimes}$.

The result is strengthened by the following theorem.

Proposition 7 *[67, 68] Let $\mathbf{A}_i$, $i = 1, \ldots, n$ be normal and $\mathbf{A}_i$ fulfill Ruspini condition (6.1.3). Furthermore, let $\blacktriangle$ be an arbitrary t-norm and let $* \leq \otimes$. Then $\mathbf{R}^{\oplus}_{\blacktriangle}$ is a solution to (6.2.1).*

PROOF:

The proof uses the same technique as the proof of Theorem 15 and therefore it is omitted. □

Proposition 7 allows us to deal with a t-norm weaker or equal to the Łukasiewicz one only in the inference mechanism but the interpretation of a fuzzy rule base can be build w.r.t. an arbitrary t-norm $\blacktriangle$. Similarly to the system of FREs with inf$\rightarrow_*$ composition, only normality of antecedents and the Ruspini condition were assumed.

6.2.4 Illustration

This subsection is devoted to an illustration of results introduced in Subsections 6.2.2 and 6.2.3.

To visually illustrate how the additive interpretations look like let us introduce the following example.

Example 6 *Let us consider pairs $(\mathbf{A}_i, \mathbf{F}_i)$, $i = 1, \dots, 9$ of fuzzy sets on $\mathcal{D}$ and $\mathcal{D}'$, respectively. Fuzzy sets $\mathbf{A}_i$ (see Figure 6.1(a)) are triangular and form a uniform fuzzy partition of $\mathcal{D} = [0,1]$. Fuzzy sets $\mathbf{F}_i$ (see Figure 6.1(b)) are triangular with kernel points equal to precise solution to equation $y = x^2$ for x being the kernel point of the corresponding $\mathbf{A}_i$ and together fulfilling the Ruspini condition.*

Distinct interpretations of the fuzzy rule base are depicted in Figure 8.3.

To illustrate the impact of the results of this section to the to possible solutions of FREs, let us consider the following example.

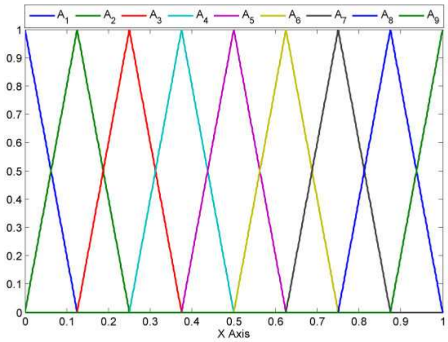
Example 7 *[67] Let there is a fuzzy rule base (1.2.4) where the antecedents are represented by fuzzy sets $\mathbf{A}_i \mathrel{\underset{\sim}{\subset}} \mathcal{D}$ which are normal and fulfill the Ruspini condition*

$$\sum_{i=1}^{n} \mathbf{A}_i(x) = 1, \quad \forall x \in \mathcal{D} \tag{6.2.4}$$

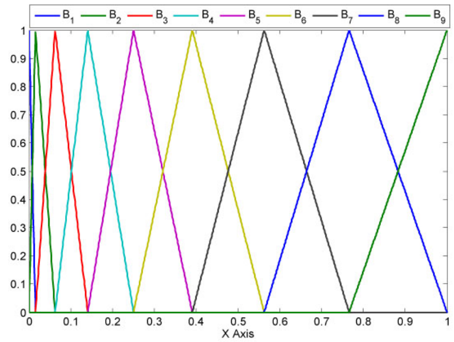
and the consequents are represented by arbitrary fuzzy sets $\mathbf{F}_i \mathrel{\underset{\sim}{\subset}} \mathcal{D}'$.

Then due to Theorem 14, fuzzy relation

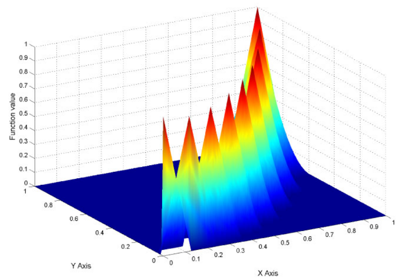
$$\mathbf{R}_{\otimes}^{\oplus}(x, y) = \bigoplus_{i=1}^{n} (\mathbf{A}_i(x) \otimes \mathbf{F}_i(y)) \tag{6.2.5}$$



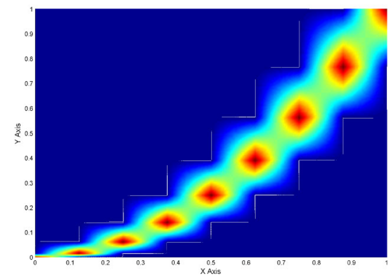
(a) Antecedent fuzzy sets



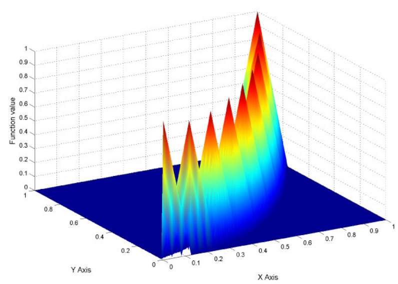
(b) Consequent fuzzy sets



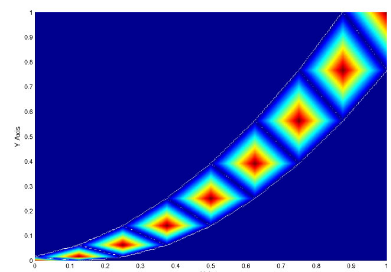
(c) Additive interpretation $\mathbf{R}_{\odot}^{\oplus}$



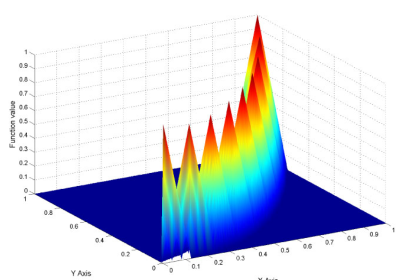
(d) Additive interpretation $\mathbf{R}_{\odot}^{\oplus}$ - view from above



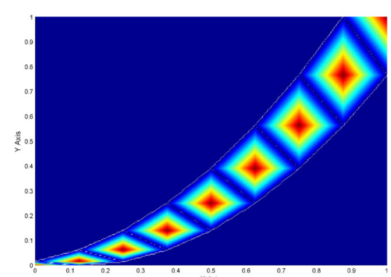
(e) Additive interpretation $\mathbf{R}_{\otimes}^{\oplus}$



(f) Additive interpretation $\mathbf{R}_{\otimes}^{\oplus}$ - view from above



(g) Disjunctive interpretation $\check{\mathbf{R}}_{\otimes}$



(h) Disjunctive interpretation $\check{\mathbf{R}}_{\otimes}$ - view from above

Figure 6.1: Different interpretations of a fuzzy rule base.

is a solution to the following system of fuzzy relation equations

$$\mathbf{A_i} \lhd_{\otimes} \mathbf{R} = \mathbf{F}_i \quad i = 1, \ldots, n,$$

as well as fuzzy relations

$$\check{\mathbf{R}}_{\otimes}(x, y) = \bigvee_{i=1}^{n} (\mathbf{A}_i(x) \otimes \mathbf{F}_i(y)) \quad \textit{and}$$

$$\mathbf{R}_{\odot}^{\oplus}(x, y) = \bigoplus_{i=1}^{n} (\mathbf{A}_i(x) \odot \mathbf{F}_i(y)) = \sum_{i=1}^{n} \mathbf{A}_i(x)\mathbf{F}_i(y)$$

due to Corollary 11 and Proposition 6, respectively.

Moreover, due to Theorem 15, fuzzy relation (6.2.5) is also a solution to the following system of fuzzy relation equations

$$\mathbf{A_i} \circ_{\otimes} \mathbf{R} = \mathbf{F}_i \quad i = 1, \ldots, n, \tag{6.2.6}$$

and due to Proposition 7, a solution to system (6.2.6) can be also found in the form of $\mathbf{R}_{\odot}^{\oplus}$.

It means that (using the terminology of [60]) fuzzy relations $\check{\mathbf{R}}_{\otimes}$, $\mathbf{R}_{\otimes}^{\oplus}$ and $\mathbf{R}_{\odot}^{\oplus}$ are *safe models* of fuzzy rules (1.2.4) if connected to the inference mechanism based on the inf$\rightarrow_{\otimes}$ composition. The two latter fuzzy relations, $\mathbf{R}_{\otimes}^{\oplus}$ and $\mathbf{R}_{\odot}^{\oplus}$ are safe even in case of the inference mechanism based on the sup-$\otimes$ composition.

It is worth noticing that fulfilling two conditions such as the Ruspini one and the normality of fuzzy sets, which is very often required in practice, led to the solvability of the corresponding fuzzy relation equations. These results are of a high practical importance since they put assumptions only on the antecedent fuzzy sets. This enables us to identify a fuzzy rule base in such a way to ensure the solvability of an adjoint system of FREs only by fuzzy partitioning the input domain $\mathcal{D}$. The consequent fuzzy sets could be arbitrary, e.g. identified by some algorithm from data which are at disposal, if any.

6.3 Fuzzy Transform of Fuzzy Relations

Fuzzy relation $\mathbf{F} : \mathcal{D} \times \mathcal{D}' \to [0,1]$ can be viewed as a fuzzy set-valued function $\mathbf{F} : \mathcal{D} \to [0,1]^{\mathcal{D}'}$, i.e., as a mapping which assigns a fuzzy subset of $\mathcal{D}'$ to each node $x \in \mathcal{D}$. In the latter, we will not distinguish between both points of view since they will be always clear from the context.

Fuzzy relation is a crucial mathematical notion related to FRB systems. Instead of a crisp control function $f : \mathcal{D} \to \mathcal{D}'$ we deal with some $\mathbf{F} : \mathcal{D} \to [0,1]^{\mathcal{D}'}$ interpreting an FRB. Therefore, approximation of fuzzy relations has the same motivation as the approximation of functions. In this section, we adapt the F-transform technique for fuzzy relations. Similarly to Chapter 2, we also consider the approximated object, fuzzy relation in this case, to be given only at some nodes $p_1, \ldots, p_N \in \mathcal{D}$.

6.3.1 Fuzzy Transform

The original ideas of the F-transform can be directly extended for fuzzy relations [69], which leads to the following formulas for the F-transform and its inversion. Let us again fix the domain $\mathcal{D} = [a, b]$ for the whole section.

Definition 41 Let a fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \underset{\sim}{\subset} \mathcal{D}, n > 2$ and let $\mathbf{F} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ be a continuous fuzzy relation. The n-tuple $[\mathbf{F}_1, \ldots, \mathbf{F}_n]$ of fuzzy sets on $\mathcal{D}'$ given by

$$\mathbf{F}_i(y) = \frac{\int_a^b \mathbf{F}(x,y)\mathbf{A}_i(x)dx}{\int_a^b \mathbf{A}_i(x)dx},\ i = 1, \ldots, n \tag{6.3.1}$$

is the *direct F-transform* of $\mathbf{F}$ with respect to the given fuzzy partition. $\mathbf{F}_1, \ldots, \mathbf{F}_n$ are the *components* of the F-transform of $\mathbf{F}$.

Definition 42 Let $[\mathbf{F}_1, \ldots, \mathbf{F}_n]$ be the direct F-transform of $\mathbf{F} \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ with respect to $\mathbf{A}_1, \ldots, \mathbf{A}_n \underset{\sim}{\subset} \mathcal{D}$. Then the fuzzy relation $\mathbf{F}_n^F \underset{\sim}{\subset} \mathcal{D} \times \mathcal{D}'$ given by

$$\mathbf{F}_n^F(x, y) = \sum_{i=1}^{n} (\mathbf{A}_i(x) \cdot \mathbf{F}_i(y)), \tag{6.3.2}$$

is be called the *inverse F-transform* $\mathbf{F}$.

Remark 22 *It may be considered questionable to transform a fuzzy relation according to formulas (6.3.1) and (6.3.1) because fuzzy relation is a function with two variables and formulas for the F-transform of a two-dimensional function (3.1.2) and (3.2.9) have been already introduced in Chapter 3. However, note that* $\mathbf{F}_n^F \equiv \mathbf{R}_{\odot}^{\oplus}$ *and so Definitions 41 and 42 introduce the F-transform in the light of FRB systems, additive interpretations and consequently again additive normal forms.*

Thus for numerical purposes the F-transform from Chapter 3 should be used but for FRB applications the last two definitions bring some advantages that will be demonstrated in the oncoming text. Moreover, this approach was motivated and later on justified by a real fuzzy control application [70, 71].

Analogously to Chapter 2, we can define a discrete version of the F-transform. An approximated fuzzy relation is seen as a fuzzy set valued function and we assume its partial knowledge i.e. that that the fuzzy relation is given at nodes $p_1, \ldots, p_N \in \mathcal{D}$.

The discrete version of the F-transform of a fuzzy relation is then defined as follows.

Definition 43 Let a fuzzy partition of $\mathcal{D}$ be given by basic functions $\mathbf{A}_1, \ldots, \mathbf{A}_n \underset{\sim}{\subset} \mathcal{D}, n > 2$ and let $\mathbf{F} : \mathcal{D} \to [0, 1]^{\mathcal{D}'}$ be a fuzzy set valued function given at nodes $p_1, \ldots, p_N \in \mathcal{D}$ such that for each $i = 1, \ldots, n$, there exists $k = 1, \ldots, N$ such that $\mathbf{A}_i(p_k) > 0$. The n-tuple $[\mathbf{F}_1, \ldots, \mathbf{F}_n]$ of fuzzy sets on $\mathcal{D}'$ given by

$$\mathbf{F}_i(y) = \frac{\sum_{k=1}^{N} \mathbf{F}(p_k, y) \mathbf{A}_i(p_k)}{\sum_{k=1}^{N} \mathbf{A}_i(p_k)}, \ i = 1, \ldots, n \tag{6.3.3}$$

is the *discrete direct F-transform* of $\mathbf{F}$ with respect to the given fuzzy partition.

Remark 23 *A generalization to more-dimensional case is straightforward and therefore it is omitted, see Chapter 3.*

The set of data $(p_k, \mathbf{F}(p_k, \cdot)) \underset{\sim}{\subset} \mathcal{D}'$ where $p_k \in \mathcal{D}$, $\mathbf{F}(p_k, \cdot) \underset{\sim}{\subset} \mathcal{D}'$ and $k = 1, \ldots, N$ can be either obtained by asking an expert how to control a given process at node p_k while $\mathbf{F}(p_k, \cdot)$ is a fuzzy sets representing a linguistic expression the expert used in his answer. For instance: *What to do when the right-hand side wall is 5cm close? Answer: turn the wheel very much to the right.* Here $p_k = 5$ and $\mathbf{F}(p_k, \cdot)$ is a fuzzy set representing the expression *very much* in the context $\mathcal{D}'$.

The second possibility is to, say during a manual control, collect nodes $(p_k, f(p_k))$ and to "fuzzify" control actions $f(p_k)$. This increases the robustness since manual control actions are typical but never precise. Further reasons for this approach will be discussed later in the text, see Chapter 7.

Chapter 7

Application to a Fuzzy Control of an Autonomous Mobile Robot

This section focuses on a possible applications of the F-transform technique to the fuzzy control. It directly uses the definitions of the F-transform of a fuzzy relation and the heuristically proposed approach introduced above.

7.1 Approaches to Identification of an FRB

In general, there are two main approaches to an identification of an FRB controlling a given process [21]. FRB systems were motivated by dealing with an expert knowledge which is more or less always expressed by a natural language where the phenomenon of vagueness is essential. However for some processes, an expert knowledge acquisition is not a trivial task or transformation of such knowledge into an FRB is technically hardly feasible [1]. In these cases a data-driven approach can be used either to *adapt* some initial very rough FRB or to *generate* a new one if no initial FRB is attainable. In general, data-driven approaches (neural learning, heuristic algorithms, adaptation optimizing a cost function etc.) deal with some training data obtained by experiments.

In the case of a generation of an FRB we just approximate given data, but it might be done by any classical approach and thus the usage of *fuzzy* is questionable although it surely provides undoubtable advantages e.g. interpretability, transparency or robustness. In the second case we assume that we are given some initial FBR that is to be adapted. However, an adaptation algorithm can lead to something completely different from the initial FRB and therefore it might be set randomly. Thus we either do not have a prior expert knowledge or we lose it, at least partially.

Finally, in any type of learning, there is a problem that the system must *learn* all possible situation otherwise the system will not be able to behave correctly. This can lead to a huge mass of experiments and even this may not be sufficient.

Here, we recall [70] an identification strategy that tries to deal with the mentioned problems and involves the F-transform technique. It can be algorithmically described as follows:

- Manually control a given process and collect input-output pairs of data $(p_k, f(p_k))$
- "Fuzzify" the collected control actions $f(p_k)$ to get pairs $(p_k, \mathbf{F}(p_k, \cdot))$
- Construct a fuzzy partition of $\mathcal{D}$
- Compute components of the extended direct F-transform
- Proceed experiments with an automatic control by fuzzy rules (1.2.4) interpreted by fuzzy relation $\mathbf{R}_{\odot}^{\oplus}$ i.e. by the extended inverse F-transform (with an appropriate defuzzification e.g. COG).

At this moment an expert only observes a behaviour of the controlled process, i.e. correctness of the model controlling the process. For those situation which were not learned sufficiently (or were not learned at all), the expert specifies an appropriate control action by linguistic expression. It means that originally collected data $(p_k, f(p_k))$ for $k = 1, \ldots, N$ were modified to $(p_k, \mathbf{F}(p_k, \cdot))$ and finally enriched by data $(p_{N+l}, \mathbf{F}(p_{N+l}, \cdot))$, $l = 1, \ldots, M$ with fuzzy control actions given linguistically by an expert. It means that the fuzzy sets $(p_{N+l}, \mathbf{F}(p_{N+l}, \cdot))$ represent appropriate evaluating linguistic expression [50, 81, 82, 83] (*small, very big, roughly five, etc.*) Some of these fuzzy sets related to evaluating linguistic expressions are depicted in Figure 7.1.

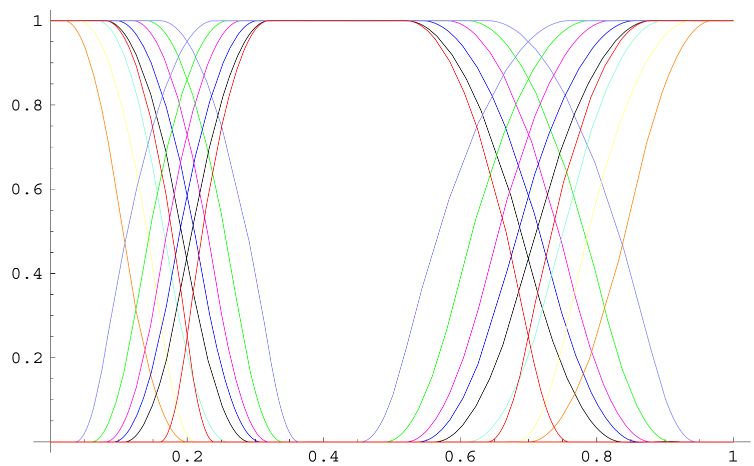


Figure 7.1: Fuzzy sets representing typical evaluating linguistic expressions.

Then the F-transform components, serving us as consequent fuzzy sets, are recomputed from all data $(p_k, \mathbf{F}(p_k, \cdot))$ where $k = 1, \ldots, N + M$ i.e. according to the following formula

$$\mathbf{F}_i(y) = \frac{\sum_{k=1}^{N+M} \mathbf{A}_i(p_k)\mathbf{F}(p_k, y)}{\sum_{k=1}^{N+M} \mathbf{A}_i(p_k)}. \tag{7.1.1}$$

This means that the data set is enriched by the expert knowledge and the original FRB generated by the F-transform is modified (recomputed). The F-transform components now aggregate both types of information - experimental and expert -

to get a well-tuned FRB system.

This approach has been successfully applied to an approximation based control [70] of an autonomous dynamic robot, see Figure 7.2.

7.2 Corridor Problem and the Robot Description

We deal with a dynamic robot with a pivoted ultrasound sensor providing partial (very imprecise) information about a robot's position in a corridor. The robot's task is to pass through any chosen corridor. The given task is vague and any effort to obtain a precise mathematical formulation of the task could lead either to something that is far from original human understanding of driving vehicles or to technically difficult systems like, e.g., systems of partial differential equations of higher orders etc. Therefore, we consider this problem to be a typical problem for an implementation of an FRB system.

Our task is to implement an automatic mechanism of cornering into the dynamic robot (see Fig. 7.2) such that the robot could move through any possible corridor. Of course, such corridors which have no sense for the robot (e.g. slimmer than the robot) are out of our scope. At the first stage, the problem of a simple corridor with no crossing and no obstacle is solved. The mechanism should be realized in a form of fuzzy controller, i.e. a controller working with an FRB consisting of n fuzzy rules.

Let us describe the main features and properties of the robot. Briefly, the robot could be described as a square metal plane with three wheels: a front wheel that is free (movable) and two back ones; see Fig. 7.2. The back wheels are powered by their own electric engines. This allows the robot to turn the wheels with different speeds. There is an incremental sensor at each back wheel measuring a number of its revolutions. It gives us information about a distance covered by the robot.

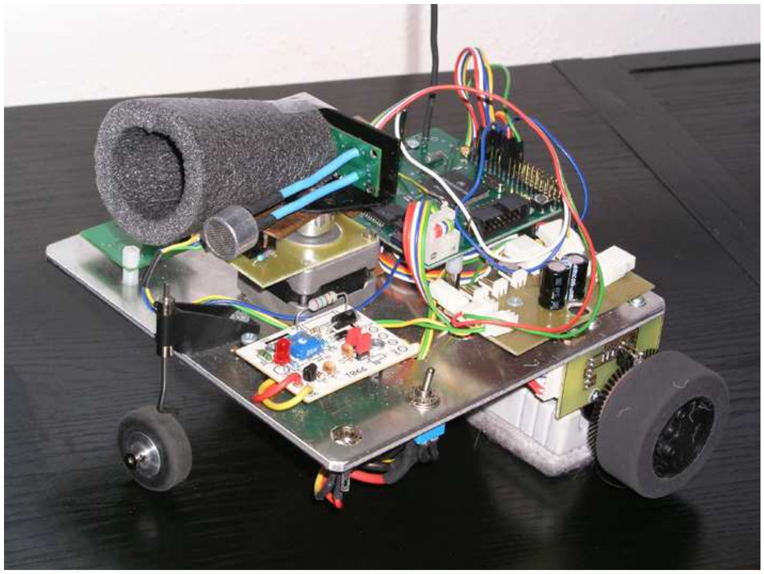


Figure 7.2: The dynamic robot controlled by the F-transform.

At the top of the robot, there is a pivoted ultrasound sensor measuring a distance to a corridor wall in a direction of the sensor. The robot measures a distance d_k on one hand side and, while moving through the corridor, the sensor rotates to measure a distance d_{k+1} on the other hand side. This gives the robot an approximate information about its position between walls on both sides.

An approximate relative distance to the middle of the corridor at a moment k is denoted by e_k and computed as

$$e_k = (-1)^{k+1} \left(\frac{2d_k}{d_k + d_{k-1}} - 1 \right) \quad k = 2, \ldots, N \tag{7.2.1}$$

where d_0 is obtained at the beginning of the robot journey when the robot takes a survey to both sides. Let us stress that the sign $-$ represents the robot's position on the left-hand side of the corridor while the sign $+$ reversely represents the right-hand side position.

Remark 24 *Note that formula (7.2.1) neglects the fact, that there was some non-zero time period between measuring the distance d_{k-1} on one side and the distance d_k on the second side while the robot was still moving. This imprecision could have been solved by having two sensors in parallel, each measuring only the distance on its own side. However, this would be a hardware solution which was not at disposal. Thus, formula (7.2.1) was another source of high imprecision in input information which favored an FRB approach.*

The relative distance e_k is the first input variable of the fuzzy controller. The second input variable is a change of this relative distance Δe_k defined as

$$\Delta e_k = e_k - e_{k-1}. \tag{7.2.2}$$

The mentioned input variables are the only input variables and hence, we have a double-input-single-output system. The output variable could be either a turning radius u or its change in time Δu. The first choice leads to a PD fuzzy controller:

$$\text{IF } e \text{ is } \mathcal{A}_i^e \text{ AND } \Delta e \text{ is } \mathcal{A}_i^{\Delta e} \text{ THEN } u \text{ is } \mathcal{F}_i, \tag{7.2.3}$$

while the second one leads to a PI fuzzy controller:

$$\text{IF } e \text{ is } \mathcal{A}_i^e \text{ AND } \Delta e \text{ is } \mathcal{A}_i^{\Delta e} \text{ THEN } \Delta u \text{ is } \mathcal{F}_i \tag{7.2.4}$$

where $i = 1, \ldots, n$ and $\mathcal{A}_i, \mathcal{F}_i$ are evaluating linguistic expressions.

7.3 Construction of the Controller

The choice of input and output variables, decisions about the type of fuzzy controller, interpretation of an FRB, shapes of fuzzy sets representing used linguistic evaluating expressions, used operations as well as identification of an FRB are the essential steps leading to the final controller. This section briefly describes all of these items.

Type of the Controller: PI v. PD

Although a PI controller was expected to be appropriate its implementation has not been successful. The problem was hidden in technical deficits of the robot. Particularly, there was a too long time period between the moments we measured the distances d_{j-1} and d_j. During that period the robot was still changing the turning radius what brought it into such a situation that it could not get out of there with inference based on rules where just a change of turning radius Δu is used as the output variable.

This does not mean that a PI controller is inappropriate in general. A similar robot equipped by at least two sensors could be a very appropriate benchmark for testing control methods given by a PI controller. Based on this, only a PD controller was successfully implemented in practice.

Fuzzy Control Methods

The inverse F-transform of a fuzzy relation produces fuzzy relation $\mathbf{F}_n^F \underset{\sim}{\subset} \mathcal{D}_e \times \mathcal{D}_{\Delta e} \times \mathcal{D}_u$ which will serve us as an interpretation of an FRB controlling the robot movement. The input to the controller is pair of a crisp numbers, say $(e', \Delta e') \in \mathcal{D}_e \times \mathcal{D}_{\Delta e}$ which is fuzzified by a singleton fuzzifier so, the output of an inference mechanism is a fuzzy set on $\mathcal{D}_u$ given by $\mathbf{F}_n^F(e', \Delta e', u)$ which is then defuzzified by the *center of gravity*.

Besides the F-transform, the *perception based logical deduction* [47, 51] was decided to be implemented to have another fuzzy model for a comparison. It is a specific linguistically-based method which considers an FRB to be a list of independent implicative rules i.e. not aggregated by the minimum operation. There is a special *pre-selection* algorithm [22] involved in the method, searching for the most appropriate rule(s) which is (are) the only fired rule(s). Antecedents and consequents are evaluating linguistic expressions [50] modelled by fuzzy sets such as depicted in Figure 7.1. The implications between antecedents and consequents are modelled by the Łukasiewicz residuation operation. Inferred fuzzy sets are then defuzzified by the *defuzzification of evaluating expressions* [51].

This choice has been made for several reasons. First of all, the perception based logical deduction is purely a linguistic method which allows an expert approach but on the other hand there exists a related learning algorithm [5, 24]. Moreover, the perception based logical deduction as well as the learning algorithm are already implemented in a complex software package *LFLC2000* [23] for fuzzy modelling which is developed by the Institute for Research and Applications of Fuzzy Modeling and therefore its licence is at disposal.

Parameters

Contexts (i.e. universes $\mathcal{D}_e, \mathcal{D}_{\Delta e}$ and $\mathcal{D}_u$) of the relative distance e, the change of the relative distance Δe and the control action u have been based on technical aspects of the robot set up as displayed in Tab 7.1

$\mathcal{D}_e$	$\mathcal{D}_{\Delta e}$	$\mathcal{D}_u$
$[-0.8, 0.8]$	[-1,1]	[-900,900]

Table 7.1: Contexts of particular linguistic variables.

Both back wheels can turn with a speed which can be set up by software from 0 up to 1600. The velocity is set to keep the sum of speed of both back wheels to be equal to 900. The control action u determines the difference of the left-hand wheel speed and the right-hand wheel speed.

Remark 25 *The speed domain* $[0, 1600]$ *of each wheel is given by the producer of the robot and by the robot component suppliers. For our task, it is not very important to go into hardware and software details.*

Identification of Fuzzy Rule Bases

The linguistic fuzzy rule base connected to the perception based logical deduction was built two different ways. The first way was purely an expert linguistic proposal, while the second way was generated from a set of training data $(e_k, \Delta e_k, u_k)$ $k = 1, \ldots N$ collected while manually driving the robot through different corridors, see Tab 10.2.

The same data were used for a construction of an FRB consisting of fuzzy rules (1.2.4) with the additive interpretation with help of the F-transform. Input domain $\mathcal{D}_e \times \mathcal{D}_{\Delta e}$ has been partitioned by a uniform fuzzy partition consisting of 5 triangular fuzzy sets on both axes. The consequent fuzzy set were determined as the components of the F-transform (6.3.3).

7.4 Results and Comparison

Perception Based Logical Deduction & Expert Identification

The first approach based on the expert construction of a fuzzy rule base (7.2.3) and the perception based logical deduction led to a successful automatic control.

Behaviour of the robot controlled by the first fuzzy rule base *PBLD1* composed of 37 rules (see Table 10.5) could be described as a save behaviour since it was good enough to avoid accidents. But the behaviour was not very smooth and the robot was oscillating too much. The smoothness was partially improved (see video files PBLD1-attempt1.avi and PBLD1-attempt2.avi) by the *smooth DEE* defuzzification method [51] - a method derived from the DEE and implemented in LFLC2000.

The second fuzzy rule base *PBLD2* composed of 43 rules (see Table 10.6) was created by modifications of some rules in *PBLD1* and adding new ones. The smoothness of the robot behaviour improved slightly (see video file PBLD2-attempt1.avi), on the other the it was not so save anymore, i.e., some accidents were recorded (see video file PBLD2-attempt2.avi).

Finally, *PBLD2* was again expertly modified to get fuzzy rule base *PBLD3* composed of 51 rules (see Table 10.7). Behaviour of the robot controlled by this FRB (see video file PBLD3) was slightly smoother that the one we got by *PBLD1* and slightly more save than the one we got by *PBLD2*.

Fuzzy Rule Bases Identified from Data

During experiments made by an expert training data Tab 10.2 were obtained. These data were used to automatically identify a fuzzy rule base.

Fuzzy rule base *PBLD-learning* identified by the linguistic learning [5, 24] algorithm implemented in the LFLC2000 is listed in Table 10.8. Since the training data do not fully cover all possible situations, the fuzzy rule base was not able to control the robot sufficiently (see vide file PBLD-learn.avi).

The same holds for the fuzzy rule base with an additive interpretation which was identified by the direct F-transform although this approach led to a smoother behaviour.

The main problem is that the training data do not contain “critical situations” since an expert manually driving the robot during experiments did not get to such situations. But instead of continuing in experiments and collecting data an automatic control of the robot by the generated FRBs was tested and observed. Situations when the automatic control did not performed well were recorded and appropriate control actions for them expertly proposed as proposed in Section 7.1.

New 3-tuples $(e_{N+l}, \Delta e_{N+l}, \mathbf{F}(e_{N+l}, \Delta e_{N+l}, \cdot))$ where $l = 1 \ldots, r$ were appended to the data set and the F-transform components recomputed according to (7.1.1). In our case, $r = 6$ additional items displayed in Table 10.3 were sufficient to get a well-tuned automatic control system, see video files FTR-attempt1.avi and FTR-attempt2.avi.

Remark 26 *Different rule bases with the additive interpretation were generated to find an appropriate compromise between complexity, robustness and precision. Five uniform triangular basic functions on both input axes were found to be appropriate.*

Analogously, FRB *PBLD-learning* was extended by 6 new rules, see Table 10.9. This extended FRB performed safely enough but comparably a bit less smooth that the one identified by the F-transform, see video files PBLD-learn-improved-attempt1.avi and PBLD-learn-improved-attempt2.avi.

Overview

Let us briefly discuss the proposed approach. What is the main aim of Definition 41 and Definition 42 besides the fact that components are fuzzy sets instead of real numbers and the inverse F-transform formula coincides with the formula for the additive interpretation of an FRB?

The main aim of the definition of the F-transform of a fuzzy relation is that the technique is "shape dependent" compared to the original one and allows a user to deal with training data as well as with an expert knowledge in a linguistic form. As such, it provides the possibility to aggregate both information into one FRB.

In Section 7.1, some typical problems which can appear during an identification have been discussed. The purely expert approach has been presented as being difficult and time-consuming, which has been approved by our experiments. On the other hand, this approach led to a successful model. The data-driven generation of an FRB requires many experiments and therefore it has been omitted from our focus. The data-driven modifications of some initial rule bases would first require some expert rule base proposal which is, again, not trivial. Moreover this proposal can be then modified to something very different.

This Section experimentally justified the suggested approach where the initial rule base is data-driven generated and then expertly improved. The experiments demonstrated that this approach is not time-consuming, as it requires a short time to collect training data and additional expert proposals improving behaviour in particular situations seem to be more consistent than a pure expert rule base construction. This approach was used to identify two FRBs - the one using the F-transform and the linguistic one. The linguistic FRB is clearly easier to interpret, however, it required 72 rules plus 6 additional rules. The approach using the F-transform generates as many rules as required by a user, in our case only 25. Moreover, this number does not increase when appending additional expert data!

Chapter 8

Neural Network Approach to the F-Transform

In this Chapter, we consider $f : \mathcal{D} \to \mathbb{R}$ to be a continuous function which is given at a finite set of nodes $p_k \in \mathcal{D}$ where $k = 1, \dots N$. The task is to approximate the given data.

Approximation is a very often studied notion so, we can hardly avoid building bridges between fuzzy approximation and already known results aiming at the approximation problem. Conversely, we can inherit many results from other branches dealing with the approximation problem.

This chapter is an introduction to the study of different relationships between a particular fuzzy approximation method (F-transform) and neural networks as another soft computing area which has been demonstrated many times to be an appropriate tool for approximation tasks.

By studying both approaches together we expect:

- development of new algorithms (known in neural networks) for fuzzy approximation
- enriching both branches by already done results from each other

- possible improvements

- answering natural question about similarities and similar problems in both branches

- inheriting theoretical results e.g. conditions of universal approximations etc.

At this first stage of our investigation, we attempted to look at the fuzzy transform from a neural network point of view to open this problematic, inherit neural algorithms, investigate possible improvements, implement an incremental type of learning and build a bridge between both branches for further results and algorithmic improvements. Good behaviour of the proposed algorithm is then experimentally justified.

8.1 RBF ϕ Neural Networks

The approximation of a function is a typical problem solved by neural networks. Compared to fuzzy techniques, neural nets are usually implemented as black boxes but they also have advantages like, an algorithmic approach to an identification of a model or *incremental* (on-line) learning algorithms. This section is devoted to the so-called ϕ-neural nets, see [41]. Basically, ϕ-neural nets are one hidden layer nets with only one linear unit (with an identity activation function) in the output layer.

Basic functions $\mathbf{A}_i$, $i = 1, \dots, n$ partitioning the domain $\mathcal{D}$ can be viewed (in the neural network terminology) as local units. Therefore, the F-transform technique is closely related to the so-called RBF (*Radial Basis Function*) neural networks which deal with the local units and we will consider only RBF ϕ-neural nets in the latter. Obviously, there exists a neural network performing the F-transform approximation i.e. the inverse F-transform, see Figure 8.1.

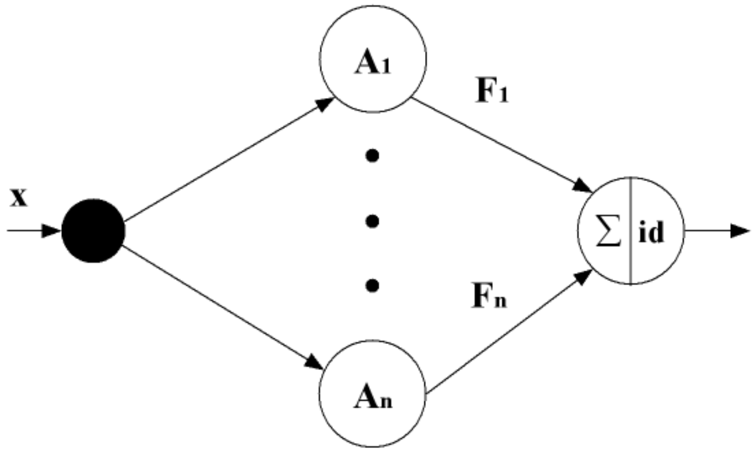


Figure 8.1: RBF ϕ-neural network performing the F-transform

There are different definitions and approaches to local unit activation functions or radial basis functions. In most practical situations, the Gaussian functions are used [26, 63] but definitions of RBF are usually more general [66].

The most usual approach to RBF units is as follows:

- The activation function is basically a continuous non-increasing function $\mathbf{A} : \mathbb{R}^+ \to [0, 1]$ (conversely to the perceptron neural nets where we require non-decreasing activation function, see [66]);
- The inner potential (conversely to the perceptron neural nets) is not computed as a weighted sum of inputs but according to the following formula

$$\xi(\mathbf{x}) = \frac{||\mathbf{x} - \mathbf{c}||}{h} \tag{8.1.1}$$

  where $\mathbf{x} \in \mathbb{R}^r$ is an input vector, $\mathbf{c} \in \mathbb{R}^r$ is a vector determining a *center* of the unit and finally $h \in \mathbb{R}^+$ is a parameter determining the width of the unit, see [8, 66].

In the latter, we restrict our focus on the case $r = 1$ for a simplified visualization.

Such an RBF network performing the inverse F-transform and constructed according to [8, 66] is similar to the one on Figure 8.1, with a few differences. First, all hidden layer units will provide the same activation function **A**. Second, the input to the i-th hidden layer unit will be marked by the weight parameter c_i determining the center of the unit. Third, each hidden layer unit will have a bias h_i determining the width parameter of the unit. Fourth, the inner potential $\xi_i : \mathbb{R} \to \mathbb{R}^+$ of the i-th unit is computed according to (8.1.1) i.e.

$$\xi_i(x) = \frac{|x - c_i|}{h_i}, \tag{8.1.2}$$

see Figure 8.2.

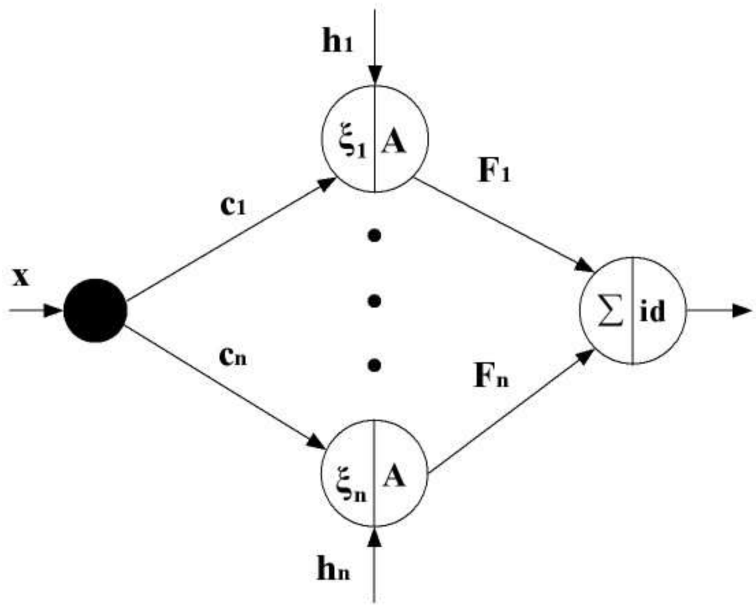


Figure 8.2: RBF ϕ-neural network

It is easy to see that the basic functions from Definition 19 can be constructed in the presented RBF neural network way. For instance, if we take $\mathbf{A}(z) = (1 - z) \wedge 0,\ z \in \mathbb{R}$ then it is easy to check that

$$\mathbf{A}_i(x) = \mathbf{A}\left(\xi_i(x)\right) \tag{8.1.3}$$

where $\mathbf{A}_i$, $i = 1, \ldots, n$ are triangular shaped basic functions determining a uniform fuzzy partition i.e. $h = h_i$ for $i = 1, \ldots, n-1$. Analogously, if we take

$$\mathbf{A}(z) = \begin{cases} \frac{1}{2}\left(\cos\left(\Pi z\right) + 1\right) & z \leq 1, \\ 0 & \text{otherwise} \end{cases} \tag{8.1.4}$$

which is for $z \in \mathbb{R}^+$ a non-increasing function then one can again check that equality (8.1.3) holds for sinusoidal shaped basic functions $\mathbf{A}_i$, $i = 1, \ldots, n$.

8.2 Fuzzy transform as a Neural Network

As already discussed, both methods - RBF ϕ-neural networks and the F-transform - provide us with an approximation of a function and are closely related to each other. Moreover, there exists an RBF neural net performing the inverse F-transform. However, the most important feature of neural networks is hidden in the possibility to learn or tune their parameters, very often incrementally.

8.2.1 Learning Algorithm

In the terminology of the neural nets, the computation of the components of the F-transform F_i according to (2.2.6) is called *off-line* (or batch) learning. However, for certain applications incremental learning algorithms have to be used, especially for on-line identification problems where we have to avoid complete rebuilding of a model because of new measurements which could yield high computational efforts.

From the original definitions we keep only the inverse F-transform formula which is performed by the RBF neural net displayed on Figure 8.1 and criterion (2.2.16) which is to be minimized. Formula (2.2.6) which is forced by the minimization of criterion (2.2.16) will be replaced by an on-line algorithm.

The most usual way how to construct an on-line learning is to consider it in the *delta rule* i.e. weights are modified by some Δ after each new sample $(p_k, f(p_k))$ is involved. The gradient descent method is a standard tool for finding Δ.

To minimize the error function $\Phi(Q_1, \ldots, Q_n) = \sum_{k=1}^{N} \Phi_k(Q_1, \ldots, Q_n)$ where each sample error Φ_k is equal to

$$\Phi_k(Q_1, \ldots, Q_n) = \sum_{i=1}^{n} (f(p_k) - Q_i)^2 \mathbf{A}_i(p_k), \quad Q_i \in \mathbb{R}, \tag{8.2.1}$$

we differentiate

$$\frac{\partial \Phi_k}{\partial Q_i}, \quad i = 1, \ldots, n \tag{8.2.2}$$

which leads to

$$\frac{\partial \Phi_k}{\partial Q_i} = 2\mathbf{A}\left(\xi_i(p_k)\right)\left(f(p_k) - Q_i\right) \tag{8.2.3}$$

where

$$\xi_i(p_k) = \frac{|p_k - c_i|}{h}. \tag{8.2.4}$$

The gradient vector has the same directions as the vector of the the fastest growth of the function values and therefore we use the negative gradient in the construction of the delta rule. Hence the delta rule is as follows

$$F_i^{(k)} = F_i^{(k-1)} + \theta_1(f(p_k) - F_i^{(k-1)})\mathbf{A}\left(\xi_i(p_k)\right) \tag{8.2.5}$$

where $0 \leq \theta_1 \leq 1$ is a learning coefficient and $F_i^{(k)}$ is the i-th component of the F-transform after k samples involved where $k = 1, \ldots, N$.

Remark 27 *Notice, that although we use standard RBF neural network and standard neural tools like the gradient descent method together with the delta rule, the error function which is minimized is different compared to usual approaches. We do not compare function values $f(p_k)$ with the outputs of the network but with its weights $F_i^{(k-1)}$. This is a significant difference which is inherited from the F-transform to keep its properties.*

8.2.2 Unsupervised Learning

In the previous subsection, we have introduced a relationship between the F-transform method and the RBF neural networks and proposed the gradient descent method for a learning algorithm. Besides the learning of the weights F_i we can get more from the neural network approach.

The construction of the basic functions can be the key issue for the results of the approximation. In general, one can hardly expect that the uniform distribution of the basic functions of the same length would provide us with best results but on the other hand, the basic functions cannot be chosen arbitrarily and some say fuzzy cluster analysis would have to be used. Therefore, in most applications, the uniform fuzzy partition has been chosen. We will discuss the possibility of the neural approach to the fuzzy partition construction.

Let us consider basic functions $\mathbf{A}_i$ comprising a fuzzy partition with a symmetry. These basic functions are functions of one variable x and three parameters c_{i-1}, c_i, c_{i+1}. Therefore, in the latter, we will again use the notation from the F-transform since it is shorter:

$$\mathbf{A}_i(x) = \mathbf{A}(x, c_{i-1}, c_i, c_{i+1}) \tag{8.2.6}$$

for $i = 1, \ldots, n$.

For instance, the triangular shaped basic functions forming a fuzzy partition with a symmetry are given by

$$\mathbf{A}_i(x) = \begin{cases} \frac{(x-c_{i-1})}{c_i-c_{i-1}} x \in [c_{i-1}, c_i] \\ \frac{(c_{i+1}-x)}{c_{i+1}-c_i} x \in [c_i, c_{i+1}] \\ 0 \quad \text{otherwise} \end{cases} \tag{8.2.7}$$

where $i = 0, \ldots, n+1$ and $c_0 = c_1,\ c_{n+1} = c_n$, while the sinusoidal shaped basic functions are given by

$$\mathbf{A}_i(x) = \begin{cases} \frac{1}{2}\left(\cos\left(\frac{\Pi(x-c_i)}{(c_i-c_{i-1})}\right)+1\right) x \in [c_{i-1}, c_i] \\ \frac{1}{2}\left(\cos\left(\frac{\Pi(x-c_i)}{(c_{i+1}-c_i)}\right)+1\right) x \in [c_i, c_{i+1}] \\ \quad 0 \quad \text{otherwise} \end{cases} \tag{8.2.8}$$

where $i = 0, \ldots, n+1$ and $c_0 = c_1,\ c_{n+1} = c_n$.

For a given shape of basic functions the centroids c_i for $i = 1, \ldots, n$ already completely specify the fuzzy partition. The task is to tune the centroids. Again, let us use the advantage of incremental self-organizing (unsupervised) algorithms already developed for neural networks, see [26, 41]. We adopt a *c-means clustering* for RBF neural networks published, e.g., in [66].

The resulting algorithm using both the self-organizing method for determining a distribution of the nodes c_i and the gradient descent method for adapting the components F_i will be as follows.

Algorithm:

(8.2.9)

FOR $k := 1$ **TO** N **DO BEGIN**

$j = \texttt{arg}\min_{i=1,\ldots,n}\{|p_k - c_i^{(k-1)}|\}$;

FOR $i := 1$ **TO** n **DO BEGIN**

IF $i = j$ **AND** $j \notin \{1, n\}$ **THEN**

$c_i^{(k)} := c_i^{(k-1)} + \theta_2(p_k - c_i^{(k-1)})$

ELSE

$c_i^{(k)} := c_i^{(k-1)}$;

$F_i^{(k)} = F_i^{(k-1)} + \theta_1(f(p_k) - F_i^{(k)})\mathbf{A}_i(p_k)$;

END;

END.

The inputs $F_i^{(0)}$ for $i = 1, \dots, n$ to the algorithm described above are small random numbers and $c_i^{(0)}$ for $i = 0, \dots, n+1$ are distributed equidistantly on the domain and keeping the conditions $c_0^{(0)} = c_1^{(0)} = a$ and $c_n^{(0)} = c_{n+1}^{(0)} = b$.

The algorithm is independent on the shape of the basic functions. In the first part, it searches for the closest centroid to an actual incoming value p_k. The chosen centroid is then shifted unless it is a corner centroid $c_1^{(k-1)}$ or $c_n^{(k-1)}$. Then the delta rule formula is applied to each component of the F-transform but because of the influence of the basic function $\mathbf{A}_i$ weighting the formula only two neighboring components are modified.

8.3 Experiments

Let us consider function f given by

$$f(x) = 2e^{(-40(x-0.5)^2)} - 1 \tag{8.3.1}$$

on a domain $[a, b] = [0, 1]$. Function (8.3.1) has been sampled to get a training set $(p_k, f(p_k))$ at randomly chosen nodes $p_k \in [a, b]$ where $k = 1, \dots, N = 100$, see Tab 10.4. For simplicity, only one learning coefficient $\theta = \theta_1 = \theta_2$ has been considered.

Obviously, incremental learning given by (8.2.5) cannot reach the accuracy obtained in case when the components are given by original formula (2.2.6). The components given by the delta rule only tend to the optimal ones given by (2.2.6).

On the other hand, through the resulting algorithm (8.2.9), which also modifies the distribution of the nodes c_i besides the components, significantly better results have been achieved. It is impossible to measure the accuracy of the approximations by criterion (2.2.16) since particular errors are weighted by the basic functions which are different for both approximations. Thus, the *simple normed least square criterion*

$$Error = 100\frac{1}{N}\sum_{k=1}^{N}\frac{(\hat{f}(p_k)-f(p_k))^2}{(\max f(p_k)-\min f(p_k))} \tag{8.3.2}$$

where $\hat{f}$ is the approximate output, have been used to measure the accuracy.

The proposed neural approach very often provided even better results than the original batch formula. For instance, for $n = 10$ the original approach gives results with 0.523 error for the triangular shaped basic functions and 0.462 for the sinusoidal shaped basic functions. The neural approach gives always different errors depending on random generation of $F_i^{(0)}$ and the choice of θ but in general, oscillating between 0.457 and 0.966 depending on different θ coefficient or methods varying the learning coefficient.

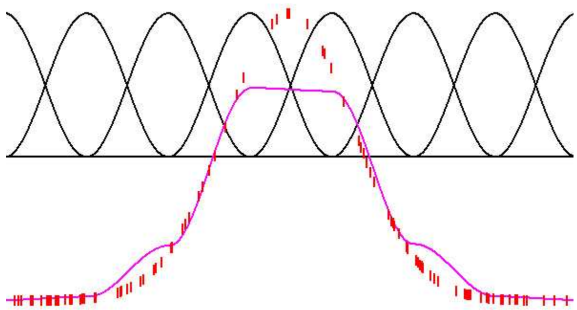
The advantage of shifting the centroid will play the more important role the less basic functions we use. For the case of $n = 7$ and sinusoidal shaped basic functions, in which the original batch formula gives error 1.227, the neural approach returns much better results, see Table 8.1.

Learning coefficient θ	**Error**
0.6	0.669
0.7	0.618
0.8	0.586

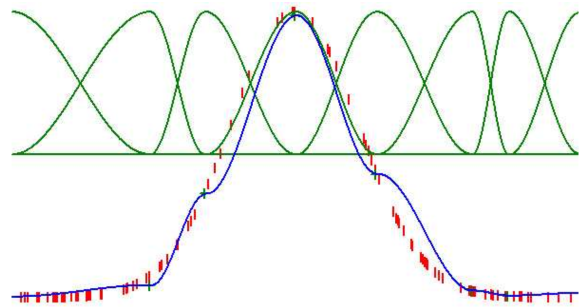
Table 8.1: Table of experimentally obtained errors of the proposed neural approach, $n = 7$.

Similar result were obtained for other combinations of number n and θ. In general, it can be stated that for smaller numbers of the basic functions the advantage of the neural network approach of shifting the centroids can compensate the higher impreciseness caused by the incremental character of the algorithm.

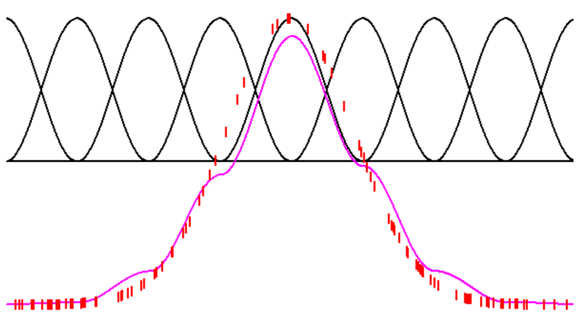
Due to Lemma 7 from [54] we can use the F-transform technique to very fast



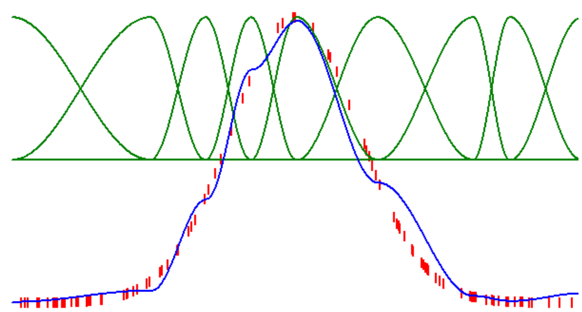
(a) Original F-transform, uniform fuzzy partition comprised from $n = 8$ sinusoidal shaped basic functions.



(b) Proposed neural improvement of the F-transform, fuzzy partition comprised from $n = 8$ sinusoidal shaped basic functions.



(c) Original F-transform, uniform fuzzy partition comprised from $n = 9$ sinusoidal shaped basic functions.



(d) Proposed neural improvement of the F-transform, fuzzy partition comprised from $n = 8$ sinusoidal shaped basic functions.

Figure 8.3: Samples of function f given by (8.3.1) and its approximations by the inverse F-transform and by the proposed neural improvement of the fuzzy transform with learning coefficient $\theta = 0.8$.

and computationally simple numerical evaluation of the definite integral of f. Due to Lemma 8, we can analogously use the F-transform w.r.t. non-uniform fuzzy partitions so, even the incremental variant with centroids shifting can be considered.

Again, function (8.3.1) has been considered. Its definite integral is according to adaptive recursive Simpson's method implemented in MATLAB® [28] equals to -0.440 (with the *tolerance responsiveness toll* $= 10^{-6}$ certifying a very high accuracy, for details see [28]). The F-transform gives numerical definite integral equal to -0.455 for $n = 10$ sinusoidal shaped basic functions and equal to -0.479 for $n = 7$ sinusoidal shaped basic functions. The neural improvement of the F-transform returns again always a bit different integral because of random start setting but of a very high preciseness, see Table 8.2.

Remark 28 *Note, that the numerical integral of function f has been computed by MATLAB using its analytical description (8.3.1) while the F-transform either in batch or neural incremental version used only a set of* 100 *random samples.*

Learning coefficient θ	**MATLAB**	**F-transform**	**Neural F-transform**
0.6	-0.440	-0.479	in (-0.459, -0.457)
0.7	-0.440	-0.479	in (-0.456, -0.455)
0.8	-0.440	-0.479	in (-0.454, -0.454)

Table 8.2: Numerical integrals by MATLAB, the fuzzy transform and the proposed neural approach to the fuzzy transform, $n = 7$.

Even for the case of $n = 10$ sinusoidal shaped basic functions, the neural algorithm (8.2.9) gives again numerical integral based on Lemma 8 which is always closer to the value -0.440, than the integral computed with help of the original formula for the fuzzy transform and based on Lemma 7, see Table 8.3.

Learning coefficient θ	**MATLAB**	**F-transform**	**Neural F-transform**
0.6	-0.440	-0.455	in (-0.447, -0.445)
0.7	-0.440	-0.455	in (-0.442, -0.441)
0.8	-0.440	-0.455	in (-0.438, -0.437)

Table 8.3: Numerical integrals by MATLAB, the fuzzy transform and the proposed neural approach to the fuzzy transform, $n = 10$.

It can be stated that the advantage of shifted centroids is even stronger in numerical integration regardless of whether the approximation was obtained incrementally.

Chapter 9

Conclusion

The main goal of the thesis was to investigate fuzzy transform - a particular fuzzy approximation method - from theoretical point of view as well as from the point of view of possible practical applications. Fuzzy approximation is a very important and well motivated field. It stems from works aiming at approximation abilities of fuzzy rule based systems.

Fuzzy transform was proposed as a method consisting of two transformations. The direct one transforming any function, continuous on a given closed interval, to a discrete vector of average values describing local information about the transformed function. The inverse one transforming the vector back to the space of continuous domain on the given interval. It turned out that the F-transform can be very powerful in different types of problems, e.g., function approximation or numerical solutions of ordinary differential equations.

The main aim of the thesis is that the F-transform was investigated from analytical and numerical point of view as well as it was studied in the context of other fuzzy approximation techniques and finally in the context neural networks - another soft computing branch very often focusing on the approximation problem.

At first, main definitions and results were recalled. Then the F-transform was elaborated even for non-uniform fuzzy partitions of the given interval. It was showed that if a very natural property - symmetry of basic functions forming a fuzzy partition - is kept than we can still get result analogous to those obtained in the case of uniform fuzzy partitions, see Chapter 2.

At second, functions with two or more variables were considered and the F-transform generalized. All the important results from the one-dimensional case were obtained even for the case of the F-transform of a function with arbitrary finite number of variables, see Chapter 3.

In Chapter 4, the F-transform was applied to numerical solutions of partial differential equations. Three main types of partial differential equations (the equations of mathematical physics) were considered and numerically solved. We applied the direct F-transform on both sides of a given differential equation to get a system of algebraic equations which could be solved by an existing method. The obtained numerical solution was transformed back by the inverse F-transform to get a continuous approximation of the respective analytical solution.

Error estimations, i.e. the fact that the numerical solutions equal to the precise analytical ones up to a certain accuracy, were proved for all three types of equations. Moreover, the convergence of the continuous solution given by the direct F-transform the numerical one, was proved for all types of equations as well. Finally, this Chapter provided readers with an example demonstrating the advantage of the proposed approach involving the F-transform method in comparison with the finite difference method.

In Chapter 5, we established the basis for further investigations of approximating abilities of normal forms - another fuzzy approximation concept. Such approach to the approximation of extensional fuzzy sets brings a new view on this field and simplifies further exploration of its properties. The main goal of this Chapter lies in introducing the F-transform in a generalized version as an eminent part of a larger group of formulas called normal forms.

At first, we proved the conditional equivalence of disjunctive and conjunctive normal forms with new, and better, estimations. At second, motivated by work [59], we established a family of additive normal forms. At third, the conditional equivalence of additive normal forms was partially clarified.

Disjunctive and conjunctive normal forms closely relate to two standard (disjunctive and conjunctive) interpretations of fuzzy rules. Based on the promising investigation of the additive normal forms, a new (additive) interpretation of a fuzzy rule base was proposed. This interpretation was also motivated by Takagi-Sugeno rules and it has a direct link to the F-transform. Chapter 6 studied the additive interpretations from approximation point of view as well as from interpolation point of view. The latter means that we elaborated conditions under which the additive interpretation of a fuzzy rule base is a a solution to adjoint systems of fuzzy relation equations, see Theorems 14 and 15 as well as Propositions 6 and 7. Moreover, we the sufficient conditions are of a very high practical importance because of their easy to fulfill character. The results were presented on illustrative examples. The Chapter also provided new definition of the F-transform of a fuzzy relation which was motivated by a real application.

Chapter 7 follows the previous one and deals with the F-transform of a fuzzy relation, i.e. studies the method in the context of fuzzy rule based systems. Our approach is motivated and demonstrated by a real application - the fuzzy control of an autonomous robot. The application is described at large and the proposed approach is justified by tests and comparisons with another fuzzy rule based approach. Practical realization of the fuzzy control application required high efforts in software and development and in overcoming technological drawbacks which finally led to a successful control tool. Video outputs of the practical tests are attached on the enclosed compact disc.

The F-transform method is finally, in Chapter 8 studied in the context of neural networks. The main idea of the Chapter is not to compare both approaches, but to enrich the fuzzy approach by advantages of the neural one. It is showed that there exists a neural network performing the inverse F-transform function. The main advantage of the neural network field is that it has well established algorithms for learning, i.e. for automatic structure and parameter determination from given data. This fact was used and the new variant of F-transform using on-line learning was proposed. It also enables to deal with non-uniform fuzzy partitions consisting of basic functions with the symmetry, for which theoretical results were obtained already in Chapter 2.

The proposed incremental improvement of the F-transform provided very promising result in approximation and made possible to be applied in fast in-line processes. Moreover, the method can serve as an “automatic” numerical integration method computing the definite integral of a function only from a given data set, say from a set of measured samples.

Most of the theoretical results were used in further results and/or in practical applications. Successful applications and implementations published in the thesis proved itself that the F-transform is a method of a high importance which is possible to be applied in distinct practical problems.

Chapter 10

Appendices

Table 10.1: List of basic evaluating linguistic expressions with their abbreviations and list of modifiers with their abbreviations. Each evaluating linguistic expressions is composed of a modifier and a basic expression. Expression “Zero” is interpreted by a fuzzy number and only “Empty” modifier can be connected to the expression.

Evaluating Linguistic Expressions			
Basic Expression	**Abbreviation**	**Modifier**	**Abbreviation**
Small	sm	Extremely	ex
Medium	me	Significantly	si
Big	bi	Very	ve
Zero	ze	Rather	ra
		“Empty”	
		More or less	ml
		Roughly	ro
		Quite roughly	qr
		Very roughly	vr

Table 10.2: Training data for an automatic generation of fuzzy rule bases serving for the dynamic robot control from Chapter 7.

Training Data for Fuzzy Control			
k	e_k	Δe_k	u_k
	$(-0.8, 0.8)$	$(-1, 1)$	$(-900, 900)$
1	0,293372412	-0,071725689	-512
2	0,271326899	0,022045513	-344
3	0,166681578	0,10464532	-56
4	0,117908893	0,048772685	-80
5	0,081348479	0,036560414	-104
6	0,514312007	-0,432963527	-464
7	0,298691589	0,215620418	-464
8	0,084977238	0,213714351	-440
9	0,050257056	0,034720182	-152
10	-0,127413127	0,177670183	-152
11	-0,071561092	-0,055852035	-32
12	0,388258435	-0,459819527	-32
13	0,439102815	-0,05084438	-296
14	0,129026365	0,31007645	-296
15	0,149107586	-0,02008122	-296
16	-0,009484597	0,158592182	-56
17	-0,03495702	0,025472423	-56
18	-0,088033737	0,053076717	16
19	-0,118824129	0,030790391	-80
20	-0,122417695	0,003593566	-80
21	-0,045873016	-0,076544679	-80
22	0,110436074	-0,15630909	-8
23	-0,333333333	0,065713904	208
24	-0,070453803	-0,26287953	184
25	-0,009420602	-0,061033201	112
26	-0,070809749	0,061389147	16
27	-0,218454259	0,147644509	16
28	-0,216910312	-0,001543947	160
29	-0,414141414	0,197231102	208

Table 10.2: Training data for an automatic generation of fuzzy rule bases serving for the dynamic robot control from Chapter 7.

Training Data for Fuzzy Control			
k	e_k	Δe_k	u_k
	$(-0.8, 0.8)$	$(-1, 1)$	$(-900, 900)$
30	-0,394926104	-0,01921531	208
31	-0,491717377	0,096791272	208
32	-0,436469265	-0,055248111	208
33	-0,040662548	-0,395806717	208
34	-0,013128492	-0,027534057	208
35	-0,536814507	-0,463185493	0
36	-0,262967598	-0,273846909	0
37	-0,211579509	-0,051388089	480
38	-0,13215859	-0,079420919	0
39	-0,025402314	-0,106756276	72
40	-0,012931034	-0,01247128	72
41	-0,021114808	0,008183773	0
42	-0,130817458	0,109702651	-24
43	-0,451295938	0,32047848	0
44	-0,408013356	-0,043282583	168
45	-0,409071924	0,001058568	504
46	-0,323321217	-0,085750707	-24
47	0,13679424	-0,63679424	-216
48	0,39503386	-0,25823962	-360
49	0,429874107	-0,034840247	-312
50	0,189434985	0,240439122	-456
51	0,045585984	0,143849001	0
52	0,508625974	-0,46303999	-192
53	0,437739943	0,070886031	-432
54	0,405278955	0,032460988	-432
55	0,303287077	0,101991878	-432
56	0,144766595	0,158520482	-192
57	0,167349867	-0,022583272	-192
58	-0,197406234	0,364756101	-216

Table 10.2: Training data for an automatic generation of fuzzy rule bases serving for the dynamic robot control from Chapter 7.

Training Data for Fuzzy Control			
k	e_k	Δe_k	u_k
	$(-0.8, 0.8)$	$(-1, 1)$	$(-900, 900)$
59	-0,272702991	0,075296758	0
60	-0,194984326	-0,077718665	0
61	-0,146446852	-0,048537474	0
62	-0,015306723	-0,131140129	0
63	0,064807741	-0,080114464	-216
64	-0,113300493	0,178108234	-216
65	-0,082136973	-0,031163519	-24
66	0,151556157	-0,23369313	96
67	0,115533576	0,036022581	96
68	0,022906793	0,092626783	168
69	0,027370091	-0,004463298	0
70	0,055166656	-0,027796565	96
71	-0,061946903	0,117113559	144
72	-0,211174057	0,149227155	120
73	-0,113631375	-0,097542683	192
74	-0,214975653	0,101344278	192
75	-0,165549427	-0,049426225	-312
76	-0,314323988	0,148774561	528
77	-0,329340197	0,015016208	528
78	-0,190013405	-0,139326792	528
79	0,398307601	-0,011835238	-144
80	0,40146201	-0,003154409	-504
81	0,374167391	0,027294619	-480
82	0,31743525	0,056732141	0
83	0,221376281	0,096058968	72
84	-0,363159429	0,000232102	0
85	-0,363159429	0,002321017	0
86	-0,363159429	0,000232102	0
87	-0,363159429	0,002321017	0

Table 10.2: Training data for an automatic generation of fuzzy rule bases serving for the dynamic robot control from Chapter 7.

Training Data for Fuzzy Control			
k	e_k	Δe_k	u_k
	$(-0.8, 0.8)$	$(-1, 1)$	$(-900, 900)$
88	-0,455196017	0,092036589	24
89	-0,476297293	0,021101276	408
90	-0,171611678	-0,304685615	-72
91	-0,091855321	-0,079756357	-72
92	0,150476426	-0,099314687	-96
93	0,411991146	-0,030056513	-264
94	0,2	0,211991146	-456
95	0,230696925	-0,030696925	-192
96	0,124273922	0,106423003	-72
97	0,109351433	0,014922489	216
98	0,118985453	-0,00963402	216
99	0,251042535	-0,132057082	-240
100	0,376085597	-0,125043062	-264
101	0,071955145	0,042466271	-312
102	-0,333572763	0,07775663	528
103	-0,323218324	-0,010544388	576
104	-0,067938302	-0,252800216	72
105	-0,333572763	0,07775663	528
106	-0,323218324	-0,010344388	576
107	-0,067938302	-0,255200216	72
108	0,0466746	-0,114612022	72
109	0,028890834	0,017783766	-72

Table 10.3: Additional data expertly appended to Table 10.2 for an identification of a fuzzy model by the F-transform, see Chapter 7.

Additional Data for Fuzzy Control			
k	e_k	Δe_k	u_k
$(-0.8, 0.8)$	(-1,1)	(-900,900)	u
110	0,7	0,2	- ve bi
111	-0,7	-0,2	ve bi
112	0,7	-0,2	- vr bi
113	-0,7	0,2	vr bi
114	0,8	-0,12	-ra bi
115	-0,8	0,12	ra bi

Table 10.4: Data generated by random sampling of function (8.3.1) and approximated by the F-transform, see Section 8.3.

Approximated Data		
k	p_k	$f(p_k)$
1	0,20668	-0,935964503
2	0,11618	-0,994481033
3	0,26328	-0,787392419
4	0,89524	-0,99613363
5	0,3693	0,009896228
6	0,13752	-0,989564302
7	0,81596	-0,963119845
8	0,79084	-0,932144107
9	0,03132	-0,999694417
10	0,83563	-0,977912704
11	0,9442	-0,999252965
12	0,21594	-0,920695164
13	0,90964	-0,997568021
14	0,74702	-0,825809898
15	0,72158	-0,719381919
16	0,84529	-0,983022395
17	0,81109	-0,958328271

Table 10.4: Data generated by random sampling of function (8.3.1) and approximated by the F-transform, see Section 8.3.

Approximated Data		
k	p_k	$f(p_k)$
18	0,89894	-0,996562386
19	0,70665	-0,637605441
20	0,24015	-0,865707325
21	0,73392	-0,77588489
22	0,02015	-0,99979998
23	0,09633	-0,997046698
24	0,22398	-0,905043315
25	0,88423	-0,994550114
26	0,267	-0,772000723
27	0,40875	0,433450795
28	0,53099	0,924626603
29	0,07633	-0,998476437
30	0,85713	-0,987826895
31	0,0833	-0,99807419
32	0,29434	-0,631639884
33	0,91534	-0,997985019
34	0,75988	-0,865791054
35	0,02707	-0,999739619
36	0,09198	-0,997435689
37	0,59563	0,387277654
38	0,89848	-0,996511576
39	0,14096	-0,988475058
40	0,56144	0,719704092
41	0,16042	-0,980147709
42	0,7281	-0,750435335
43	0,8696	-0,991528259
44	0,55885	0,741269214
45	0,67375	-0,402149956
46	0,07947	-0,998306016
47	0,70449	-0,624500907

Table 10.4: Data generated by random sampling of function (8.3.1) and approximated by the F-transform, see Section 8.3.

Approximated Data		
k	p_k	$f(p_k)$
48	0,04852	-0,999424471
49	0,68343	-0,479369871
50	0,16164	-0,979479941
51	0,72489	-0,735489228
52	0,31861	-0,46363807
53	0,31406	-0,498323688
54	0,63558	-0,041255128
55	0,2759	-0,731709578
56	0,83456	-0,977269976
57	0,38753	0,205828816
58	0,10852	-0,995647985
59	0,98449	-0,999832759
60	0,84802	-0,984260308
61	0,62104	0,113067722
62	0,46976	0,928165199
63	0,73142	-0,765209459
64	0,46918	0,925435709
65	0,05208	-0,999345832
66	0,69098	-0,535032252
67	0,3243	-0,418225533
68	0,20639	-0,936399001
69	0,12067	-0,993670302
70	0,31386	-0,499814777
71	0,62926	0,025131889
72	0,20058	-0,944587289
73	0,1357	-0,990102098
74	0,96377	-0,999633003
75	0,35933	-0,093690822
76	0,20518	-0,938184808
77	0,29501	-0,627563536

Table 10.4: Data generated by random sampling of function (8.3.1) and approximated by the F-transform, see Section 8.3.

Approximated Data		
k	p_k	$f(p_k)$
78	0,49706	0,999308632
79	0,87388	-0,992540769
80	0,62555	0,064638261
81	0,73093	-0,763072103
82	0,68094	-0,460128518
83	0,75268	-0,844442543
84	0,29399	-0,633756777
85	0,47767	0,960504866
86	0,42024	0,550660356
87	0,34814	-0,204918937
88	0,34096	-0,272834866
89	0,80599	-0,952737969
90	0,81265	-0,959919034
91	0,24422	-0,853947503
92	0,64865	-0,173640789
93	0,5727	0,618881665
94	0,67904	-0,445154556
95	0,06724	-0,998884108
96	0,53089	0,925103046
97	0,09259	-0,997384158
98	0,50001	0,999999992
99	0,80984	-0,957014217
100	0,64189	-0,106102089

Table 10.5: Fuzzy rule base *PBLD1*, see Section 7.4 .

Fuzzy Rules							
i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$	i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$
1	-bi	+me	+ve bi	2	-bi	+sm	+ve bi
3	-bi	-sm	+bi	4	-bi	-me	+sm
5	-me	+me	+bi	6	-me	+sm	+me
7	-me	-sm	ze	8	-me	-me	-sm
9	-sm	+me	+bi	10	-sm	+sm	+me
11	-sm	-sm	ze	12	+sm	+bi	+sm
13	+me	+bi	ze	14	-sm	+bi	+me
15	-me	+bi	+bi	16	-sm	-bi	-sm
17	-me	-bi	ze	18	+sm	-bi	-me
19	+me	-bi	-bi	20	-sm	-me	-sm
21	+sm	-me	-bi	22	+sm	-sm	-me
23	+sm	+sm	ze	24	+sm	+me	+sm
25	+me	-me	-bi	26	+me	-sm	-me
27	+me	+sm	ze	28	+me	+me	+sm
29	+bi	+me	-sm	30	+bi	+sm	-bi
31	+bi	-sm	-ve bi	32	+bi	-me	-ve bi
33	ze	ze	ze	34	ze	+sm	-me
35	ze	-sm	+me	36	ze	+me	-bi
37	ze	-me	+bi				

Table 10.6: Fuzzy rule base *PBLD2*, see Section 7.4 .

Fuzzy Rules							
i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$	i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$
1	-bi	-bi	-sm	2	-bi	-me	ze
3	-bi	-sm	+sm	4	-bi	+sm	+me
5	-bi	+me	+bi	6	-bi	+bi	+ve bi
7	-me	+bi	+bi	8	-me	+me	+me
9	-me	+sm	+me	10	-me	-sm	ze
11	-me	-me	-sm	12	-me	-bi	ze

Table 10.6: Fuzzy rule base *PBLD2*, see Section 7.4 .

Fuzzy Rules							
i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$	i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$
13	-sm	+bi	+bi	14	-sm	+me	+bi
15	-sm	+sm	+me	16	-sm	-sm	ze
17	-sm	-me	-sm	18	-sm	-bi	-sm
19	+sm	-bi	-bi	20	+sm	-me	-bi
21	+sm	-sm	-me	22	+sm	+sm	ze
23	+sm	+me	+me	24	+sm	+bi	+sm
25	+me	-bi	-bi	26	+me	-me	-me
27	+me	-sm	-me	28	+me	+sm	ze
29	+me	+me	+sm	30	+me	+bi	ze
31	+bi	+bi	+sm	32	+bi	+me	ze
33	+bi	+sm	-sm	34	+bi	-sm	-me
35	+bi	-me	-bi	36	+bi	-bi	-ve bi
37	ze	ze	ze	38	ze	+bi	+bi
39	ze	+me	-bi	40	ze	+sm	-me
41	ze	-sm	+me	42	ze	-me	+bi
43	ze	-bi	-bi				

Table 10.7: Fuzzy rule base *PBLD3*, see Section 7.4 .

Fuzzy Rules							
i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$	i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$
1	-bi	-bi	-sm	2	-bi	-me	ze
3	-bi	-sm	+sm	4	-bi	ze	+me
5	-bi	+sm	+me	6	-bi	+me	+bi
7	-bi	+bi	+ve bi	8	-me	+bi	+bi
9	-me	+me	+me	10	-me	+sm	+me
11	-me	-sm	ze	12	-me	ze	+ve sm
13	-me	-me	-ve sm	14	-me	-bi	-me
15	-sm	+bi	+bi	16	-sm	+me	+me
17	-sm	+sm	+sm	18	-sm	-sm	ze

Table 10.7: Fuzzy rule base *PBLD3*, see Section 7.4 .

Fuzzy Rules							
i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$	i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$
19	-sm	ze	ze	20	-sm	-me	-sm
21	-sm	-bi	-qr sm	22	+sm	-bi	-bi
23	+sm	-me	-me	24	+sm	-sm	-sm
25	+sm	ze	ze	26	+sm	+sm	ze
27	+sm	+me	+sm	28	+sm	+bi	+qr sm
29	+me	-bi	-bi	30	+me	-me	-me
31	+me	-sm	-me	32	+me	ze	-ve sm
33	+me	+sm	ze	34	+me	+me	+ve sm
35	+me	+bi	+me	36	+bi	+bi	+sm
37	+bi	+me	ze	38	+bi	+sm	-sm
39	+bi	-sm	-me	40	+bi	ze	-me
41	+bi	-me	-bi	42	+bi	-bi	-ve bi
43	ze	ze	ze	44	ze	+bi	+me
45	ze	+me	+me	46	ze	+ro sm	+sm
47	ze	+ve sm	ze	48	ze	-ro sm	ze
49	ze	-ve sm	-sm	50	ze	-me	-me
51	ze	-bi	-me				

Table 10.8: Fuzzy rule base *PBLD-learning*, see Section 7.4 .

Fuzzy Rules							
i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$	i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$
1	ze	ze	ze	2	ml me	-ve sm	-qr bi
3	vr sm	ex sm	-me	4	ro sm	sm	-ve sm
5	ra sm	si sm	-sm	6	ro bi	-me	-vr bi
7	me	ro sm	-vr bi	8	ve sm	si sm	-ra sm
9	-ra sm	ml sm	-ra sm	10	-sm	-ve sm	-si sm
11	vr bi	-ml me	-si sm	12	ra sm	vr sm	-vr sm
13	ml sm	-ex sm	-vr sm	14	-ex sm	ra sm	-ve sm

Table 10.8: Fuzzy rule base *PBLD-learning*, see Section .
7.4

Fuzzy Rules							
i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$	i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$
15	-si sm	ex sm	-ve sm	16	-ra sm	si sm	-sm
17	-ve sm	-ve sm	-sm	18	ra sm	-ra sm	-ex sm
19	-me	ve sm	ro sm	20	-sm	-qr sm	ro sm
21	-ex sm	-ve sm	sm	22	-qr sm	ra sm	ex sm
23	-qr sm	-ex sm	ml sm	24	-vr bi	ro sm	ro sm
25	-vr bi	-ve sm	ro sm	26	-si sm	-me	ro sm
27	-ro bi	-ml me	ze	28	-vr sm	-qr sm	ze
29	-qr sm	-si sm	vr bi	30	-ra sm	sm	-ex sm
31	-qr bi	vr sm	ze	32	-vr bi	-si sm	ml sm
33	-vr bi	ex sm	qr bi	34	-me	-sm	-ex sm
35	ra sm	-ro bi	-ro sm	36	vr bi	-qr sm	-me
37	vr bi	-si sm	-ml me	38	ve sm	ra sm	ze
39	ro bi	-ml me	-ro sm	40	vr bi	ve sm	-vr bi
41	ml sm	ra sm	-ro sm	42	-qr sm	ml me	-ro sm
43	-vr sm	ve sm	ze	44	sm	-ve sm	-ro sm
45	-sm	-si sm	-ex sm	46	ml sm	-ro sm	sm
47	si sm	sm	ml sm	48	si sm	-ex sm	ze
49	-ve sm	sm	ra sm	50	-ra sm	-sm	ro sm
51	-qr sm	sm	ro sm	52	-ro sm	-si sm	-ml me
53	-me	ra sm	qr bi	54	-ro sm	-ra sm	qr bi
55	vr bi	-ex sm	-qr bi	56	ml me	si sm	-vr bi
57	me	ve sm	ze	58	qr sm	sm	ve sm
59	-ml me	ex sm	ze	60	-qr bi	sm	ex sm
61	-qr bi	ex sm	ml me	62	-ro sm	-vr sm	-ve sm
63	ml sm	-sm	-sm	64	qr sm	ro sm	-vr bi
65	qr sm	-si sm	-ro sm	66	ra sm	ex sm	ro sm
67	ra sm	-ex sm	ro sm	68	vr sm	-sm	-qr sm
69	ml me	-sm	-qr sm	70	sm	si sm	-ml me
71	-me	-ex sm	ro bi	72	ve sm	-sm	ve sm

Table 10.9: Additional fuzzy rules expertly proposed and appended to fuzzy rule base *PBLD-learning* to improve its performance , see Section 7.4

Additional Fuzzy Rules							
i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$	i	$\mathcal{A}_i^e$	$\mathcal{A}_i^{\Delta e}$	$\mathcal{F}_i$
1	ve bi	ro sm	-ve bi	2	-ve bi	- ro sm	ve bi
3	ve bi	-ro sm	-vr bi	4	-ve bi	ro sm	vr bi
5	ex bi	-sm	-ra bi	6	-ex bi	sm	ra bi

Bibliography

[1] Babuška, R. and Setnes, M.(1999), Data-driven construction of transparent fuzzy models. In: *H.B. Verbruggen and H.-J. Zimmermann and R. Babuška, (Eds.) Fuzzy Algorithms for Control*. Kluwer, Boston, Dordrecht, London, Chap. 4, 83–106.

[2] Bandler, W. and Kohout, L.J.(1980), Fuzzy power sets and fuzzy implication operators. *Fuzzy Sets and Systems*, 4, 183–190.

[3] Bandler, W. and Kohout, L.J.(1978), Fuzzy relational products and fuzzy implication operators. In: *Proc. of International Workshop on Fuzzy Reasoning Theory and Applications*, Queen Mary College, University of London.

[4] Bezdek, J.C.(1981), *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York.

[5] Bělohlávek, R. and Novák, V.(2002), Learning rule base in linguistic expert systems. *Soft Computing*, 7, 79–88.

[6] Buckley, J. and Hayashi, Y.(1993), Fuzzy input-output controllers are universal approximators. *Fuzzy Sets and Systems*, 58, 273–278.

[7] Castro, J.(1993), Fuzzy logic controllers are universal approximators. *IEEE Transactions on Systems, Man and Cybernetics*, 25, 629–635.

[8] Coufal, D.(2005), Radial implicative fuzzy systems. In: *Proceedings of the FUZZ-IEEE'05*, Nevada, Reno, 963–968.

[9] Ruan, D. and Kerre, E.E.(2000), *Fuzzy If-Then Rules in Computational Intelligence: Theory and Applications*. Kluwer, Boston.

[10] Daňková, M.(2004), Generalized extensionality of fuzzy relations. *Fuzzy Sets and Systems*, 148, 291–304.

[11] Daňková, M.(2007), On approximate reasoning with graded rules. *Fuzzy Sets and Systems*, to appear.

[12] Daňková, M.(2002), Representation of logic formulas by normal forms. *Kybernetika*, 38, 717–728.

[13] Daňková, M. and Perfilieva, I.(2003), Logical approximation II. *Soft Computing*, 7, 228–233.

[14] Daňková, M. and Štěpnička, M.(2006), Fuzzy transform as an additive normal form. *Fuzzy Sets and Systems*, 157, 1024–1035.

[15] De Baets, B.(2000), Analytical solution methods for fuzzy relation equations. In: *The Handbook of Fuzzy Set Series, Vol. 1, D. Dubois, H. Prade (Eds.)*, Academic Kluwer Publ., Boston, 291–340.

[16] De Baets, B.(1996), A note on Mamdani controllers. In: *Proc. of the 2nd International FLINS Workshop, R. Da, P. D'hondt, P. Govaerts and E. Kerre, (Eds.)*, World Scientific, Mol, Belgium, 22–28.

[17] De Baets, B. and Mesiar, R.(1998), T-partitions. *Fuzzy Sets and Systems*, 97, 211–223.

[18] Dubois, D. and Prade, H.(1984), A theorem on implication functions defined from triangular norms. *Stochastica*, 8, 267–279.

[19] Dubois, D. and Prade, H.(1996), What are fuzzy rules and how to deal with them. *Fuzzy Sets and Systems*, 84, 169-185.

[20] Dubois, D., Prade, H. and Ughetto, L.(1997). Checking the coherence and redundancy of fuzzy knowledge bases, *IEEE Transactions on Fuzzy Systems*, 5, 398–417.

[21] Dubois D., Prade H. and Ughetto L.(1999), Fuzzy logic, control enginnering and artificial intelligence. In: *Fuzzy Algorithms for Control, H.B. Verbruggen and H.-J. Zimmermann and R. Babuška, (Eds.)*, Kluwer, Boston, Dordrecht, London, 17–57.

[22] Dvořák, A.(2000), On preselection of rules in fuzzy logic deduction. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 5, 563–572.

[23] Dvořák, A., Habiballa, H., Novák, V. and Pavliska, V.(2003), The concept of LFLC 2000 its specificity, realization and power of applications. *Computers in Industry*, 51, 269–280.

[24] Dvořák, A. and Novák, V.(2002), The extraction of linguistic knowledge using fuzzy logic and generalized quantifiers. In: *Computational Intelligent Systems for Applied Research, V. Švec, D. Ruan, P. D'hondt, E. Kerre (Eds.)*, World Scientific, Singapore, 113–120.

[25] Evans, G., Blackledge, J. and Yardley, P. (2000), *Numerical Methods for Partial Differential Equations.* Springer-Verlag, London.

[26] Fuller, R.(2000) *Introduction to Neuro-fuzzy Systems*. In: J. Kacprzyk (Eds.) Series: Advances in Soft Computing, Physica-Verlag, Heidelberg.

[27] Gan, M.-T., Hanmandlu, M. and Tan, A.H.(2005), From Gaussian mixture model to additive fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 13, 303–316.

[28] Gander, W. and Gautschi, W.(2000), Adaptive quadrature - revisited. *BIT*, 40(1), 84–101.

[29] Gottwald, S. (2001), *A Treatise on Many-Valued Logics, Studies in Logic & Computation Series.* Research Studies Press, Baldock.

[30] Gottwald, S.(2000), Generalized solvability behaviour for systems of fuzzy equations. In: *Discovering the World with Fuzzy Logic, V. Novák and I. Perfilieva (Eds.), (Studies in Fuzzines and Soft Computing series)*, Springer-Verlag, Heidelberg, New York, 401–430.

[31] Gottwald, S.(2006), On a generalization of fuzzy relation equations. In: *Proc. of 11th IPMU*, Paris, 2572–2577.

[32] Hájek, P.(1998), *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht.

[33] Kaleva, O.(1987), Fuzzy differential equations. *Fuzzy Sets and Systems*, 24, 301–317.

[34] Kaleva, O.(1990), The Cauchy problem for fuzzy differential equations, *Fuzzy Sets and Systems*, 35, 389–396.

[35] Klawonn, F.(2000), Fuzzy points, fuzzy relations and fuzzy functions. In: *Discovering the World with Fuzzy Logic, V. Novák and I. Perfilieva (Eds.), (Studies*

in Fuzzines and Soft Computing series), Springer-Verlag, Heidelberg, New York 431–453.

[36] Klawonn, F. and Kruse, R.(1993), Equality relations as a basis for fuzzy control. *Fuzzy sets and systems*, 54, 147–156.

[37] Klawonn, F. and Novák, V.(1996), The relation between inference and interpolation in the framework of fuzzy systems. *Fuzzy Sets and Systems*, 81, 331–354.

[38] Klement, E.P., Mesiar, R. and Pap, E.(2000), *Triangular Norms.* Kluwer, Boston, Dordrecht.

[39] Klir, G.J. and Yaun, B.(1995), *Fuzzy Sets and Fuzzy Logic.* Prentice Hall, New Jersey.

[40] Kosko B.(1992), Fuzzy systems as universal approximators. In: *Proc. of the 1st IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'92)*, 1153–1162.

[41] Kurková, V.(2003), Approximation of functions by neural networks. In: *Artificial Intelligence vol. 4, V. Mařík, O. Štěpánková and J. Lažanský, (Eds.)*, Academia, Prague, in Czech, 254–273.

[42] Mamdani, E.H. and Assilian,S.(1975), An experiment in lingustic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7, 1–13.

[43] Mesiar, R. and Novák V.(1999), Operations fitting triangular-norm-based biresiduation. *Fuzzy Sets and Systems*, 104, 77–84.

[44] Míka, S. and Přikryl, P.(1996), *Numerické metody řešení diferenciálních rovnic - Evoluční rovnice.* Západočeská univerzita v Plzni, Vydavatelství ZČÚ, Plzeň, in Czech.

[45] Nosková, L.(2005), Systems of fuzzy relation equation with inf$\rightarrow$ composition: solvability and solutions. *Journal of Electrical Engineering*, 56(12/s), 69–72.

[46] Novák, V.(2004), On fuzzy equality and approximation in fuzzy logic. *Soft Computing*, 8, 668–675.

[47] Novák, V.(2005), Perception-based logical deduction. In: *Computational Intelligence, Theory and Applications (Advances in Soft Computing), B. Reusch (Eds.)*, Springer-Verlag, Berlin, Heidelberg, New York, 237–250.

[48] Novák V. (2000), *Základy fuzzy modelování.* Technická literatura BEN, Praha, in Czech.

[49] Novák, V. and Lehmke, S.(2006), Logical structure of fuzzy IF-THEN rules. *Fuzzy Sets and Systems*, 157, 2003-2029.

[50] Novák, V. and Perfilieva, I.(1999), Evaluating linguistic expressions and functional fuzzy theories in fuzzy logic. In: *Computing with Words in Information/Intelligent Systems 1, L.A. Zadeh, J. Kacprzyk (Eds.)*, Springer-Verlag, Heidelberg, 383–406.

[51] Novák, V. and Perfilieva, I.(2004), On the semantics of perception-based fuzzy logic deduction. *Journal of Intelligent Systems*, 19, 1007–1031.

[52] Novák, V., Perfilieva, I. and Močkoř, J.(1999), *Mathematical Prinicples of Fuzzy Logic.* Kluwer, Boston, Dordrecht, London.

[53] Perfilieva, I. and Chaldeeva, E.(2001), Fuzzy transformation and its applications. In: *4th Czech - Japan Seminar on Data Analysis and Decision Making under Uncertainity*, 116–124.

[54] Perfilieva, I.(2003), Fuzzy approach to solution of differential equations with imprecise data: application to reef growth problem. In: *Fuzzy Logic in Geology, R.V. Demicco and G.J. Klir (Eds.), Chap. 9*, Academic Press, Amsterdam, 275–300.

[55] Perfilieva, I.(2004), Fuzzy function as an approximate solution to a system of fuzzy relation equations. *Fuzzy Sets and Systems*, 147, 363-383.

[56] Perfilieva, I.(1999), Fuzzy logic normal forms for control law representation. In: *Fuzzy Algorithms for Control, H.B. Verbruggen, H.-J. Zimmermann R. Babuška, (Eds.)*, Kluwer, Boston, Dordrecht, London, 111–125.

[57] Perfilieva, I.(2004), Fuzzy transforms. In: *Transactions on Rough Sets II. Rough Sets and Fuzzy Sets, J.F. Peters and A. Skowron (Eds.), LNCS 3135*, 63–81.

[58] Perfilieva, I.(2005), Fuzzy transforms: Theory and Applications. *Fuzzy Sets and Systems*, 157, 993–1023.

[59] Perfilieva, I.(2004), Normal forms in BL and ŁΠ algebras of functions. *Soft Computing*, 8, 291–298.

[60] Perfilieva, I. and Lehmke, S.(2005), Safe modeling of fuzzy if-then rules. In: *Computational Intelligence, Theory and Applications, B. Reusch, (Eds.)*, Springer Physica-Verlag, Berlin, Heidelberg, New York, 235–240.

[61] Perfilieva, I. and Valášek, R.(2005), Fuzzy transforms in removing noise. In: *Computational Intelligence, Theory and Applications, B. Reusch, (Eds.)*, Springer Physica-Verlag, Berlin, Heidelberg, New York, 221–230.

[62] Ruspini, E.H.(1969), A new approach to clustering. *Inform. and Control*, 15, 22–32.

[63] Rutkowska, D.(2002), *Neuro-fuzzy Architectures and Hybrid Learning.* Springer-Verlag, Heidelberg, New York.

[64] Shmilovici, A. and Maimon, O.(1998), On the solution of differential equations with fuzzy spline wavelets. *Fuzzy Sets and Systems*, 96, 77–99.

[65] Smith, G.D.(1964), *Numerical Solutions of Partial Differential Equations.* Oxford University Press, London.

[66] Šíma, J. and Neruda, R.(1996), *Teoretické otázky neuronových sítí.* Matfyzpres, Prague, in Czech.

[67] Štěpnička, M., De Baets, B. and Nosková , L.(2007), Arithmetic fuzzy models. *Fuzzy Sets and Systems*, to be submitted.

[68] Štěpnička, M., De Baets, B. and Nosková , L.(2007), On Additive and multiplicative fuzzy models. In: *New Dimensions in Fuzzy Logic and Related Technologies, Vol. 2, M. Štěpnička, V. Novák, U. Bodenhofer, (Eds.)*, University of Ostrava, 95–102.

[69] Štěpnička, M. and Lehmke, S.(2004), Approximation of fuzzy functions by extended fuzzy transforms. In: *Computational Intelligence, Theory and Applications, B. Reusch, (Eds.) (Advances in Soft Computing).* Springer-Verlag, Berlin, Heidelberg, New York, 187–195.

[70] Štěpnička, M. and Valášek, R.(2005), Approximation based fuzzy control. In: *Proc. of the 8th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainity*, 131–139.

[71] Štěpnička, M. and Valášek, R.(2005), Dynamic robot control based on fuzzy approximation. *Journal of Electrical Engineering*, 56(12/s), 17–20.

[72] Štěpnička M. and Valášek R.(2004), Fuzzy transforms and their applications to heat flow equation. In: *Abstracts of the 2nd FLLL/SCCH Master and PhD Seminar 2004*, 51–56.

[73] Štěpnička, M. and Valášek, R.(2003), Fuzzy transforms for function with two variables. In: *Proc. of the 6th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainity*, 100–107.

[74] Štěpnička, M. and Valášek, R.(2004), Fuzzy transforms and their application to wave equation. *Journal of Electrical Engineering*, 55(12/s), 7-10.

[75] Štěpnička, M. and Valášek, R.(2005), Numerical solution of partial differential equations with help of fuzzy transform. In: *Proc. of the 2005 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'05)*, Reno, Nevada, 1153–1162.

[76] Takagi, T. and Sugeno, M.(1985), Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15, 116–132.

[77] Trillas, E. and Valverde, L.(1985), On implication and indistinguishability in the setting of fuzzy logic. In: *Management Decision Support Systems Using Fuzzy Sets and Possibility Theory, J. Kacprzyk and R.R. Yager, (Eds.)*, Springer Physica-Verlag, Rheinland, Koln, 198–212.

[78] Vetterlein, T. and Štěpnička, M.(2006), Completing fuzzy if-then rule bases by means of smoothing splines. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14, 235–244.

[79] Zadeh, L.A.(1973), Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. on Systems, Man and Cybernetics*, 3, 28–44.

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

[81] Zadeh, L.A.(1975), The concept of linguistic variable and its application to approximate reasoning I. *Inform. Sci.*, 8, 199–250.

[82] Zadeh, L.A.(1975), The concept of linguistic variable and its application to approximate reasoning II. *Inform. Sci.*, 8, 301–357.

[83] Zadeh, L.A.(1975), The concept of linguistic variable and its application to approximate reasoning III. *Inform. Sci.*, 9, 43–80.

[84] Zhang, Y., Wang, G. and Liu, S.(1998), Frequency domain methods for the solutions of N-order fuzzy differential equations. *Fuzzy Sets and Systems*, 94, 45–59.

List of Author's Publications

Publications in International Journals

- Daňková, M. and Štěpnička, M.(2006), Fuzzy transform as an additive normal form. *Fuzzy Sets and Systems*, 157, 1024–1035.

- Štěpnička, M.(2003), Fuzzy transformation and its applications in A/D converter. *Journal of Electrical Engineering*, 54(12/s), 72–75.

- Štěpnička, M.(2006), Inference mechanisms, systems of fuzzy relational equations and the additive interpretations of rule bases. *Journal of Electrical Engineering*, 57(7/s), 78-81.

- Štěpnička, M. and Daňková, M.(2007), the On fuzzy normal forms. *Journal of Electrical Engineering*, 58, to appear.

- Štěpnička, M. and Polakovič, O.(2007), A neural network approach to the fuzzy transform. *Fuzzy Sets and Systems*, submitted.

- Štěpnička, M. and Valášek, R.(2005), Dynamic robot control based on fuzzy approximation. *Journal of Electrical Engineering*, 56(12/s), 17–20.

- Štěpnička, M. and Valášek, R.(2004), Fuzzy transforms and their application to wave equation. *Journal of Electrical Engineering*, 55(12/s), 7-10.

- Vetterlein, T. and Štěpnička, M.(2006), Completing fuzzy if-then rule bases by means of smoothing splines. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14, 235–244.

Publications in Multi-author Volumes

- Drozd, P., Štěpnička, M., Dolný, A. and Völkl, P.(2006), Fuzzy modeling - prospective tool for conservation biology. In: *Environmental changes and biological assessment III. P. Kočárek, V. Plášek, and K. Malachová (Scripta Facultatis Rerum Naturalium Universitatis Ostraviensis Nr. 163)*, University of Ostrava, 7–14.

  Štěpnička, M., De Baets, B. and Nosková , L.(2007), On Additive and multiplicative fuzzy models. In: *New Dimensions in Fuzzy Logic and Related Technologies, Vol. 2, M. Štěpnička, V. Novák, U. Bodenhofer, (Eds.)*, University of Ostrava, 95–102.

- Štěpnička, M. and Lehmke, S.(2004), Approximation of fuzzy functions by extended fuzzy transforms. In: *Computational Intelligence, Theory and Applications, B. Reusch, (Eds.) (Advances in Soft Computing)*. Springer-Verlag, Berlin, Heidelberg, New York, 187–195.

- Štěpnička, M. and Valášek, R.(2006), Generating a fuzzy rule base with an additive interpretation. In: *Proceedings of the 16th IFAC World Congress - Volume C2 - Cognition and Control (AI, Fuzzy, Neuro, Evolut.)*, Prague, Elsevier, 233–238.

Publications in Conference Proceedings

- Bodenhofer, U., Daňková, M., Štěpnička, M. and Novák, V.(2007), A plea for the usefulness of the deductive interpretation of fuzzy rules in engineering applications. In: *Proc. of the 2007 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'07)*, 1567–1572.

- Daňková, M. and Štěpnička, M.(2005), Genetic algorithms in fuzzy approximation. In: *Proc. of the Joint 4th EUSFLAT and 11th LFA*, 651–656.

- Štěpnička, M.(2007), Fuzzy transform for practical problems. In: *Proc. of the Inteligentní systémy pro praxi 2007 (Inteligent systems for practice)*, Lázně Bohdaneč, Ostrava, AD&M, 39–40.

- Štěpnička, M., Lughofer, E. and Pavliska, V.(2006), Comparison of data-driven fuzzy modelling methods tested on NOx data. In: *Abstracts of the 2nd FLL-L/SCCH Master and PhD Seminar*, Linz, Austria, 33–39.

- Štěpnička, M. and Nosková, L.(2006), Fuzzy relation equations - new solutions and solvability criteria. In: *Proceedings of the conference Logic of Soft Computing 5 & 5th workshop of the ERCIM working group on Soft Computing*, Malaga, Spain, 61–67.

- Štěpnička, M. and Nosková, L.(2006), Systems of fuzzy relation equations: New solvability criteria based on the orthogonality condition. In *Proceedings of Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty - Ninth Meeting*, Waseda University, Japan, 352–356.

- Štěpnička, M. and Polakovič, O.(2006), Fuzzy transform from a neural network point of view. In: *Proc. of 11th IPMU*, Paris, France, 1860–1867.

- Štěpnička, M. and Valášek, R.(2005), Approximation based fuzzy control. In: *Proc. of the 8th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainity*, 131–139.
- Štěpnička M. and Valášek R.(2004), Fuzzy transforms and their applications to heat flow equation. In: *Abstracts of the 2nd FLLL/SCCH Master and PhD Seminar 2004*, 51–56.
- Štěpnička, M. and Valášek, R.(2003), Fuzzy transforms for function with two variables. In: *Proc. of the 6th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainity*, 100–107.
- Štěpnička, M. and Valášek, R.(2005), Numerical solution of partial differential equations with help of fuzzy transform. In: *Proc. of the 2005 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'05)*, Reno, Nevada, 1153–1162.

Index