Fuzzy Numerical Schemes for Hyperbolic Differential Equations

Michael Breuss[1] and Dominik Dietrich[2]

[1] Saarland University, Germany
breuss@mia.uni-saarland.de
[2] German Research Centre for Artificial Intelligence (DFKI GmbH), Bremen, Germany
dietrich@dfki.de

Abstract. The numerical solution of hyperbolic partial differential equations (PDEs) is an important topic in natural sciences and engineering. One of the main difficulties in the task stems from the need to employ several basic types of approximations that are blended in a nonlinear way. In this paper we show that fuzzy logic can be used to construct novel nonlinear blending functions. After introducing the set-up, we show by numerical experiments that the fuzzy-based schemes outperform methods based on conventional blending functions. To the knowledge of the authors, this paper represents the first work where fuzzy logic is applied for the construction of simulation schemes for PDEs.

1 Introduction

Fuzzy logic (FL) is an important tool in computer science [9,6,19] and engineering [10,18,16,7]. It allows to control complex processes based on a small number of expert rules where an explicit knowledge of the behaviour of the considered system is given. In this paper, we consider an entirely new field of application for fuzzy logic: the construction of numerical schemes for hyperbolic partial differential equations (PDEs). Such equations arise in many disciplines, e.g., in gas dynamics, acoustics, geophysics, or bio-mechanics, see e.g. [11] for an overview. They describe a time-dependent transport of a given quantity without an inherent tendency to an equilibrium.

Hyperbolic numerics. The difficulty to handle hyperbolic PDEs stems from the fact that their solutions usually involve the formation of discontinuities. In order to cope with these, modern high-resolution schemes employ a nonlinear mixture of approximation schemes. For components of such a blending, two classic candidates to which we stick in this work are the monotone upwind method and the Lax-Wendroff (LW) method that is second order accurate. Second order accurate methods such as the LW scheme give a reasonable accuracy on smooth solutions, but they fail near discontinuities. Monotone schemes such as the upwind method are always only first order accurate, yet they offer the advantage that they can deal with discontinuities. The idea behind high-resolution (HR) schemes is to take the best of these two worlds, performing a blending in such

B. Mertsching, M. Hund, and Z. Aziz (Eds.): KI 2009, LNAI 5803, pp. 419–426, 2009.

a way that a monotone scheme is used at discontinuities while a higher order method is taken at smooth solution parts. For a more detailed exposition of these topics see e.g. [11].

Fuzzy logic for high-resolution schemes. The desired blending of first and second order approximations is performed by a so-called *limiter function* completely defining the HR scheme and its approximation properties. In the design of such a limiter only a few restrictions are imposed in the hyperbolic theory. Consequently, a wide variety of limiters has been proposed in the literature, motivated by experiences with computational examples; see [17] for an overview. However, the main *expert rules* in the design of a limiter are due to the already mentioned basic idea: *(i)* to use a monotone scheme at discontinuities, and *(ii)* to use a higher order scheme in smooth solution parts. It seems natural to ask if the blending process can be formalised using FL.

Our contribution. To our best knowledge, we employ FL for the first time within the literature for the construction of numerical schemes for PDEs here by applying the concept of FL at HR schemes for hyperbolic PDEs. Moreover, our work goes much beyond a proof-of-concept: we show that the use of FL yields significant improvements in the approximation quality.

Related work. As we have clarified above, to our knowledge no work has dealt before with the area of application we consider in this paper. Thus, we cannot rely on previous work in this field.

Structure of the paper. In Paragraph 2, we briefly review the fuzzy tools we employ in this work. Paragraph 3 introduces the mathematical background on hyperbolic PDEs and flux limiter. This is followed by an exposition on the construction of fuzzy limiters in Section 4. In Paragraph 5 we show results of numerical simulations. The paper is concluded by Paragraph 6.

2 Fuzzy Systems

Fuzzy logic is derived from *fuzzy set theory* [20], where the general idea is to allow not only for full membership or no membership, represented by the membership values 1 and 0, respectively, but also for *partial membership*, represented by a membership value in the interval $[0, 1]$. Consequently, a fuzzy set F over a domain G is represented by a membership function $\mu_F : G \to [0, 1]$. Similarly, in fuzzy logic the degree of truth is not restricted to true (1) and false (0), but can take any value of the interval $[0, 1]$, which builds the foundation for approximate reasoning and fuzzy control.

A *fuzzy system* is an expert system based on fuzzy logic. It consists of four components: fuzzy rules, fuzzifier, inference engine, and defuzzifier. *Fuzzy rules* are of the form "IF A then B" where A and B are fuzzy sets. A convenient way is to represent fuzzy sets by a (set of) parameterised function. Typical examples are monotonic decreasing functions, piecewise linear functions, exponential functions, or symmetric triangular functions. Moreover, it is common to attach a name to each fuzzy set and to identify the set by its name, resulting in more natural rules. The *fuzzifier* maps discrete input values to fuzzy sets, thus activating the rules. The *inference engine* maps fuzzy input sets to fuzzy output

sets and handles the way in which rules are combined. Finally, the *defuzzifier* transforms the fuzzy output set to a discrete output value.

Fuzzy rules can either be formulated by an expert or be extracted from given data. Once all parameters have been fixed, a fuzzy system defines a nonlinear mapping, called control function, from the input domain to the output domain. For further details we refer to [7,16].

An advantage of fuzzy expert systems over other approaches is that often a small number of expert rules are sufficient to encode explicit knowledge about the problem to be controlled. This is because fuzzy reasoning adapts a rule to a given situation by modifying the conclusion of the rule, based on a difference analysis between the situation encoded in the premise and the given situation. In this sense, we can see fuzzy reasoning as a kind of knowledge based interpolation.

Additionally, the control function can easily be adapted by changing the fuzzy rules, the inference engine or the parameters of the underlying fuzzy sets, either manually, or by applying learning and optimisation techniques [1,2,12]. One particularly simple but powerful adaptation consists of applying *hedges* or *modifiers*, which are functions on fuzzy sets which change their shape while preserving their main characteristics. Within this paper we consider the standard hedges given by the *contrast operator*, the *dilation operator*, and the *concentration operator* (see [7,16] for details).

For our application, fuzzy systems seem to offer a natural framework to represent so-called flux limiter schemes, see next paragraph. Modification of the parameter results in a new control function, which corresponds to a new numerical scheme within our application. In particular it allows the use of learning and optimisation techniques in the context of hyperbolic PDEs.

3 Hyperbolic PDEs and Flux Limiter

Hyperbolic PDEs. Let us consider the usual format of a hyperbolic PDE

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}f\left(u(x,t)\right) \;=\; 0\,; \tag{1}$$

- $x \in \mathbb{R}$ and $t \in \mathbb{R}^+$ are typically interpreted as space and time variables, respectively, whereas $\partial/\partial t$ and $\partial/\partial x$ are the corresponding spatial and temporal differential operators,
- $u(x,t)$ is the unknown density function of a quantity of interest,
- f denotes the flux function that describes the flow behaviour.

The PDE (1) needs to be supplemented by an initial condition $u_0(x)$. Then, the differential equation may be understood to encode the temporal change of the beginning state u_0. Note that we only consider here the one-dimensional spatial setting, as our developments are readily extendable to higher dimensions.

Without going into details, let us remark that solving (1) is a non-trivial task, as the solution involves the formation of discontinuities even if the initial function u_0 is arbitrarily smooth [5].

The basic numerical set-up. To make up a numerical scheme, one needs to give a discrete representation of the computational domain in space and time, as well as a discrete representation of the differential operators in (1). Thus, we define a grid in space and time featuring the widths Δx and Δt, respectively. For a discrete representation of derivatives we need the approximation of $u(x,t)$ at the grid points. Indicating discrete data by large letters, we write U_j^n for the approximation of $u(x,t)$ at the point $(j\Delta x, n\Delta t) \in \mathbb{R} \times \mathbb{R}^+$.

Most numerical schemes for hyperbolic PDEs mimic the format of (1), as this proceeding ensures the validity of useful properties. For the discretisation point with indices (j,n) in space and time, they can be read as

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{F_{j+1/2}^n - F_{j-1/2}^n}{\Delta x} = 0\,, \tag{2}$$

The indices $j \pm 1/2$ indicate a *numerical flux* between the points j and $j+1$, or $j-1$ and j, respectively. The formula (2) is then evaluated for each spatial point j of the computational domain; at the end points of the necessarily finite interval in space one needs numerical boundary conditions. The process begins at the time level $n = 0$, building up a solution iterating from time level to time level until a prescribed stopping time.

The key to success in the definition of a suitable scheme is the construction of the so-called *numerical flux function* F. In our paper, we exactly elaborate on this topic, constructing via F high-resolution schemes.

Flux limiters. Generally speaking, there exist two powerful frameworks to construct HR schemes for hyperbolic PDEs, using so-called slope limiters and flux limiters, respectively; c.f. [11,17] for detailed expositions. These approaches are only roughly equivalent. The slope limiter construction is based on geometric insight making use of the slopes of higher order interpolation polynomials incorporated in such schemes. In our work, we use the flux limiter framework that relies on the physical interpretation of scheme components as fluxes of the quantity considered in the underlying PDE.

As indicated, the basic idea is to use a first order monotone scheme at discontinuities and a high order scheme in smooth solution parts. Both schemes are usually cast in the format (2), and we identify them via a monotone numerical flux $F_{j\pm1/2,M}^n$ and a high order numerical flux $F_{j\pm1/2,H}^n$, respectively.

In order to distinguish discontinuities from smooth solution parts, we also need at $j \pm 1/2$ and time level n a *smoothness measure* we denote by $\Theta_{j\pm1/2}^n$. This is usually computed making use of the ratio of consecutive slopes depending on the flow direction. In the usual basic set-up the flow is given from left to right, and the smoothness measure is computed as

$$\Theta_{j+1/2}^n := \frac{U_{j+1}^n - U_j^n}{U_j^n - U_{j-1}^n}\,. \tag{3}$$

A *limiter function* Φ is then a function evaluating Θ and assigning it a reasonable output, so that the construction idea of HR schemes is met and we can write the high-resolution numerical flux $F_{j+1/2,HR}^n$ as

$$F_{j+1/2,HR}^n = F_{j+1/2,M}^n + \Phi\left(\Theta_{j+1/2}^n\right) F_{j+1/2,H}^n\,. \tag{4}$$

Thus, HR methods with numerical fluxes as in (4) can be viewed as a monotone scheme plus a correction. Note that for $\Phi = 0$ the monotone numerical flux is obtained, whereas $\Phi = 1$ yields the high-resolution numerical flux.

One should note that there are only a few theoretical restrictions on the definition of the limiter function Φ. The most important one is concerned with ensuring the so-called *total variation (TV) stability* of the numerical scheme, which means in practice that no oscillations are generated numerically if the scheme is TV stable, c.f. [11]. For the proceeding in our work, it is important that the limiter function is in the corresponding so-called *TV region*, see next paragraph.

4 Fuzzy Limiter Construction

From an abstract point of view the control problem we consider here is to determine the flux limiter value $\Phi \in [0, 2]$, given a smoothness measure Θ. To get a TV stable method, the control function must lie in the TV-region, which is sketched in Fig. 1 as the shaded area. Modelling a solution to this control problem using fuzzy logic requires the following steps: *(i)* Defining fuzzy sets on the input and the output domain *(ii)* Choosing a suitable inference mechanism and defuzzification method, and *(iii)* Defining a suitable rule base. Rather than starting from scratch, it is a good idea to look at a simple, already existing HR scheme. The probably simplest is the *Minmod* scheme, which linearly blends from the upwind scheme to the LW scheme, as shown in Fig. 1.

One can clearly see how the expert knowledge, (i) to use LW if the solution is smooth, and (ii) to use the Upwind method if the solution is not smooth, has been translated into the behaviour of the numerical scheme. Moreover, we can also see how the scheme fits smoothly into the framework of FL: We can think of a region in which the flux limiter is constant to correspond to a situation in which a single rule fires with full activation degree; the remaining parts correspond to situations in which several rules are active (to some degree) and their result combined by means of FL.

As we have identified two key situations in the case of the Minmod scheme, we define two fuzzy sets "smooth" and "extremum" on the input parameter, which we call "smoothness". Similarly, we define two (singleton) fuzzy sets "UP" (corresponding to a flux limiter value of 0) and "LW" (corresponding to a flux limiter value 1) on the output parameter, which we call "limiter". Our modelling

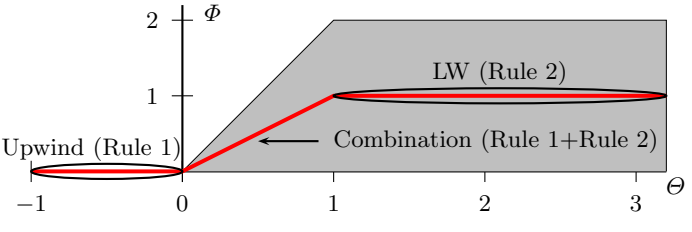


Fig. 1. TV-region and control function of the Minmod scheme

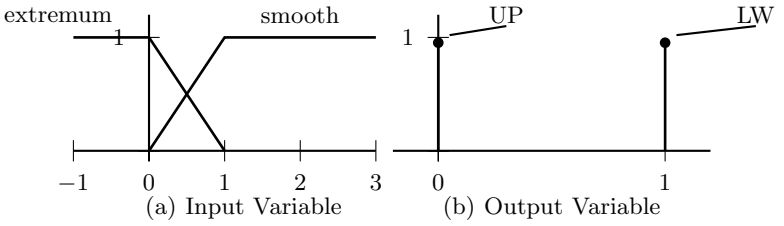


Fig. 2. Fuzzy sets for the Minmod limiter

is shown in Fig. 2. This allows us to explicitly express the construction idea of the Minmod numerical scheme in form of the following rule base:

If **smoothness** is **extremum** then **limiter** is UP
If **smoothness** is **smooth** then **limiter** is LW

Choosing an inference engine consisting of the min operator as implication operator, the max operator as aggregation operator, and the centroid method for defuzzification, we exactly obtain the control function of the Minmod scheme.

Similarly, other prominent TV-scheme such as the Superbee scheme or the MC scheme (see [11] for details) can easily be modelled within the FL framework.

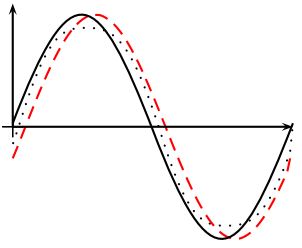
5 Experimental Study

In addition to providing a common framework and embedding existing schemes into it, new numerical schemes can be constructed by modifying the parameters within of the FL setting. This is a common step in FL design, often called *parameter tuning*. As this is usually a time consuming task, complex learning techniques have been developed to automate it [1,2,12]. Instead of relying on these we will follow the more pragmatic approach of modifying a numerical scheme only by applying the fuzzy modifiers concentration, dilation, contrast to the input fuzzy sets. The benefits are (i) simplicity and (ii) the guarantee that only the blending between the specified key situations is modified.

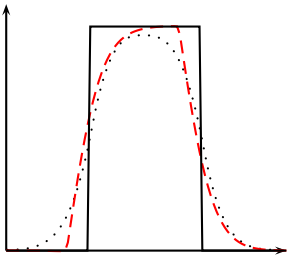
To evaluate our approach we consider the *linear advection equation*, which is probably the simplest hyperbolic PDE and which is commonly used as benchmark problem. It describes the transport of a quantity within a medium, e.g., the advection of some fluid through a pipe. In the one dimensional case it is given by the PDE

$$u_t + \overline{a} u_x = 0. \tag{5}$$

Employing periodic boundary conditions, we can construct a situation where our initial condition has moved by the PDE (5) once over the complete domain, i.e., ideally it should match again the initial. This is a typical test in the hyperbolic field. We show the results of corresponding numerical experiments for the parameters $\Delta t = 0.0025$, $\Delta x = 0.01$.



(a) Original and modified Minmod scheme



(b) Original and modified Minmod scheme

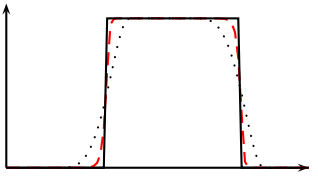
Errors sine wave:

400	0.0067	0.0041	38%
800	0.0127	0.0076	40%
2000	0.0317	0.0165	47%
4000	0.0561	0.0268	52%

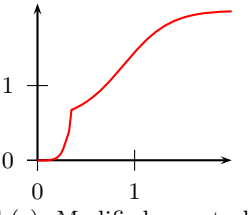
Errors box function:

400	0.0569	0.0461	18%
800	0.0725	0.0581	19%
2000	0.0993	0.0787	20%
4000	0.125	0.0992	21%

(c) Error Analysis I



(d) Original and modified scheme



(e) Modified control function MC

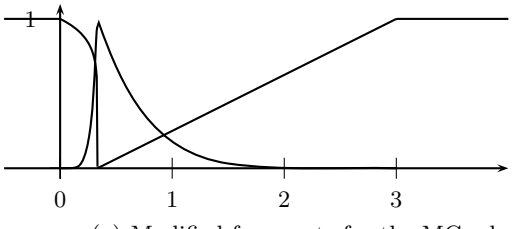
Errors sine wave:

400	0.00141	0.00121	13%
800	0.00246	0.00225	8%
2000	0.00532	0.00499	6%
4000	0.00948	0.00916	3%

Errors box function:

400	0.03239	0.00880	72%
800	0.03888	0.00884	77%
2000	0.04991	0.00900	81%
4000	0.06075	0.00934	84%

(f) Error Analysis II



(g) Modified fuzzy sets for the MC scheme

Errors sine wave:

400	0.00487	0.00350	27%
800	0.00885	0.00632	28%
2000	0.01820	0.01406	22%
4000	0.02538	0.02487	2%

Errors box function:

400	0.0176	0.0123	30%
800	0.0181	0.0124	31%
2000	0.0182	0.0127	30%
4000	0.0182	0.0131	28%

(h) Error Analysis III

Fig. 3. Results of the experimental study

For an error analysis, we give in Fig. 3(c,f,h), from left to right, *(i)* the number of iterations, *(ii)* the L_1-error of (c) the Minmod, (f) the MC, and (h) the Superbee scheme, *(iii)* the L_1-error of the corresponding Fuzzy scheme, and *(iv)* the error improvement by our approach. Let us stress, that the algorithms constructed by us perform significantly better than the original schemes.

Fig. 3(a,b,d) graphically show the difference between the original scheme (dotted), the modified scheme (red dashed line), and the reference solution (black line). Fig. 3(g) shows an example of a modified fuzzy set resulting in the control function Fig. 3(e). For more complex PDEs the improvements are similar.

6 Summary and Conclusion

For the first time in the literature, we have applied fuzzy logic for constructing numerical schemes for partial differential equations. Considering an important class of methods, namely high-resolution schemes for hyperbolic equations, we have shown that the fuzzy numerical approach results in a considerable quality gain compared with standard schemes in that field. In our future work, we aim to incorporate more sophisticated data analysis and learning strategies in our approach.

References

1. Chiang, C.K., Chung, H.Y., Lin, J.J.: A Self-Learning Fuzzy Logic Controller Using Genetic Algorithms with Reinforcements. Transaction on Fuzzy Systems 5 (1997)
2. Chin, T.C., Qi, X.M.: Genetic algorithms for learning the rule base of fuzzy logic controller. Fuzzy Sets and Systems, 1–7 (1998)
3. Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. Arch. Rat. Mech. Anal. 78, 315–333 (1982)
4. Clarke, F., Ekeland, I.: Solutions périodiques, du période donnée, des équations hamiltoniennes. Note CRAS Paris 287, 1013–1015 (1978)
5. Evans, L.C.: Partial Differential Equations. Oxford University Press, Oxford (1998)
6. Goodridge, S.G., Kay, M.G., Luo, R.C.: Multi-layered fuzzy behavior fusion for reactive control of an autonomous mobile robot. In: Procs. of the 6th IEEE Int. Conf. on Fuzzy Systems, Barcelona, SP, pp. 579–584 (1996)
7. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic, Theory and Applications (1995)
8. Harris, J.: Fuzzy Logic Applications in Engineering Science. Springer, Heidelberg (2005)
9. Kosko, B.: Fuzzy Thinking: The New Science of Fuzzy Logic, New York (1993)
10. Lee, C.C.: Fuzzy logic in control systems: Fuzzy logic controller, part I. IEEE, Trans. Fuzzy Syst., 2, 4–15 (1994)
11. LeVeque, R.J.: Finite Volume Methods for Hyperbolic Problems (Cambridge Texts in Applied Mathematics). Cambridge University Press, Cambridge (2002)
12. Lim, M.H., Rahardja, S., Gwee, B.H.: A GA paradigm for learning fuzzy rules School of EEE, Nanyang Technological University (1996)
13. Michalek, R., Tarantello, G.: Subharmonic solutions with prescribed minimal period for nonautonomous Hamiltonian systems. J. Diff. Eq. 72, 28–55 (1988)
14. Rabinowitz, P.: On subharmonic solutions of a Hamiltonian system. Comm. Pure Appl. Math. 33, 609–633 (1980)
15. Tarantello, G.: Subharmonic solutions for Hamiltonian systems via a $\mathbb{Z}_p$ pseudoindex theory. Annali di Matematica Pura (to appear)
16. Terano, T., Asai, K., Sugeno, M.: Fuzzy Systems Theory and Its Applications. Academic, New York (1992)
17. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics, 2nd edn. Springer, Heidelberg (1999)
18. Von Altrock, C.: Recent Successful Fuzzy Logic Applications in Industrial Automation IEEE. In: Fifth International Conference on Fuzzy Systems (1996)
19. Xu, D., Keller, J.M., Popescu, M.: Applications of Fuzzy Logic in Bioinformatics (Advances in Bioinformatics and Computational Biology) (2008)
20. Zadeh, L.A.: Fuzzy Sets. Information and Control (1965)