# WAI'96: II Workshop on Computer Arithmetic, Interval and Symbolic Computation

# WAI'96: Второй семинар по компьютерной арифметике, интервальным и символьным вычислениям

On August 7–8, 1996, the second Workshop on Computer Arithmetic, Interval and Symbolic Computation was held in Recife, Brazil. The main purpose of the workshop was to bring together researchers interested in scientific computation and related topics and to present and discuss recent advances in this branch of computer science and its applications.

Total about 27 talks and 10 posters were presented. 81 participants from 8 countries (Brazil, France, Germany, Mexico, Portugal, Russia, Switzerland and the USA) attended the workshop.

The major topics of interest included but were not limited to: theoretical foundations of the computational arithmetic, interval algorithms, interval arithmetic co-processors, interval probability, programming languages for scientific computation, tools for scientific computation and symbolic computation.

The workshop was held on the campus of the Federal University of Pernambuco. It was a part of the largest annual event in computer science in Brazil, the XVI Meeting of the Brazilian Computing Society. The XVI SBC, on August 4–9, 1996, attracted 1500 participants and accommodated a wide variety of events in computer science.

Marcilia A. Campos

# Interval'96: International Conference on Interval Methods and Computer Aided Proofs in Science and Engineering

## Interval'96: Международная конференция по интервальным методам и компьютерным доказательствам в науке и технике

The meeting organized by the Faculty of Mathematics and Computer Science, Würzburg University and the Editorial Board of the International Journal *Reliable Computing* was held in Würzburg from September 29 through October 2, 1996.

91 scientists from 18 countries participated in the conference. Due to the support of DFG, GAMM and other firms and institutions 22 researchers mainly from Eastern Europe were able to attend.

In 8 highlighted talks, 56 regular talks and 8 poster presentations various aspects of the topic of the conference were considered.

- Mathematical foundations and generalisations of interval arithmetic, its embedding in programming languages and libraries as well as applications in different technical fields.
- Accurate, reliable calculation of functions, new numerical methods for point and interval problems as well as verification of mathematical facts.
- Functional, formal specification of interval mathematics, complexity issues as well as coupling of interval procedures with methods from Constraint (Logic) Programming.

The following highlighted talks were presented:

- F. Benhamou (Université d'Orlèans): *A Fixpoint Approximate Semantics for Cooperating Numerical Solvers.*
- V. Gehrke (RWTH Aachen): *Direct Computation of All Singular Points of Chemical Engineering Models Using Interval Methods.*
- P. Van Hentenryck (Brown University): *Helios: a Modeling Language for Global Optimization Using Interval Analysis.*
- E. Hyvönen (VTT Helsinki): *Interval Constraint Satisfaction—a New Computational Basis for Spreadsheet Computations.*
- M. Mrozek (Jagiellonian University, Krakow): *Topological Methods in Computer Assisted Proofs in Dynamics.*
- M. Nakao (Kyushu University, Fukuoka): *A Computational Approach for the Estimates of Constants Related to the Numerical Verification Method for Boundary Value Problems.*

- J. Rohn (Charles University, Prague): *Complexity of Some Linear Problems with Interval Data.*
- S. Shary (ICT Novosibirsk): *Algebraic Approach in the Outer Problem for Interval Linear Equations.*

A selection of the talks will be published as a special issue of the journal *Reliable Computing*. The volume of abstracts of the conference can be ordered from the Editorial Board.

The meeting continued a series of conferences which has been started in 1992 near Moscow with Interval'92 (Interval and Stochastic Methods in Science and Engineering) and in 1994 in St.Petersburg with Interval'94 (Interval and Computer Algebraic Methods in Science and Engineering). This series will be continued in 1998 probably in Nanjing, China or Sozopol, Bulgaria. A possible topic may be the connection to fuzzy logic.

J. Wolff von Gudenberg
Chairman of Interval'96

# Interval Talks at FUZZ-IEEE'96

# Интервальные доклады на FUZZ-IEEE'96

The annual IEEE International Conference on Fuzzy Systems (FUZZ–IEEE) is one of the largest meetings of fuzzy researchers.- This year, FUZZ–IEEE was organized in New Orleans, on September 8–11. More than 300 papers from all over the world were presented during the conference.

## 1. Intervals: general relation to fuzzy, Zadeh's vision

Many fuzzy researchers use interval arithmetic, for two main reasons:

- First, a *fuzzy property* of a real number (like "young," "big," etc.) can be described by different intervals that correspond to different "degrees of belief": e.g., "hot" can be described as, say, $[40, 100]$ with degree of belief 1, $[35, 100]$ with degree of belief 0.9, etc.
- Second, the degrees of belief themselves are known only approximately, and in many cases, it is more natural to represent them not by a real number, but rather by an *interval.*

Due to this connection, most textbooks on fuzzy logic contain chapters on interval methods, and most fuzzy conferences have several interval-related papers presented. This conference was no exception.

Intervals were first mentioned in the very first talk, by Lotfi Zadeh, the founder of fuzzy logic [14]. In his talk, Zadeh promoted what he called *granularity*: When we measure or estimate a physical quantity, we usually have a very approximate idea of its value: e.g., if we measure, we get the value with a very low accuracy; when an expert estimates this quantity, he often can tell only whether it is "big" or "small" or "medium." In other words, we have a *low* granularity. However, when we represent this knowledge in the computer, we have to use the datatype that is currently hardware supported: real numbers. Real numbers are perfect to represent *high* granularity measurements and estimates, but when we use the multi-bit real numbers to represent low granularity, low accuracy data, we waste computer memory on storing the unnecessary bits and we waste computer time on processing them. What is needed is a methodology of handling low granularity data directly. This methodology, Zadeh thinks, will come from combining the existing methods of handling low granularity data, in particular:

- interval methods developed to handle uncertainty in measurements, and
- methods from Artificial Intelligence, developed to handle granularity of expert estimates.

## 2. Voice recognition: direct illustration of Zadeh's idea

Zadeh's talk was more a challenge than a description of methods and results, but there were several talks at this conference that made Zadeh's desired "combination" seem promising. The most relevant was a paper [8] where interval methods are used to speed up *voice recognition*.

The goal of voice recognition is, given the signal that records a human speech (i.e., its intensity at different moments of time), to determine what exactly was said. Voice recognition methods traditionally use general pattern recognition algorithms. These algorithms start with a set of test inputs $(x_1^{(k)}, \ldots, x_n^{(k)})$, $1 \leq k \leq K$, for which the desired output $y^{(k)}$ is known. They are usually based on a *model* of the dependency between $x_i$ and $y$, i.e., on the formula $y = f(x_1, \ldots, x_n, c_1, \ldots, c_p)$ with fixed $f$ and arbitrary parameters $c_j$. For example:

- we can have a *linear* model in which $p = n + 1$ and
$$f(x_1, \ldots, x_n, c_1, \ldots, c_p) = c_1 \cdot x_1 + \cdots + c_n \cdot x_n + c_{n+1}$$
- we can have a more general *polynomial* model, in which the dependency $f$ is a quadratic or a cubic polynomial with coefficients $c_j$;
- we can have a non-linear model that describes the input-output relation of a neural network; in this model, the coefficients $c_j$ are called *weights*.

Then, we *train* the model, i.e., determine the values $c_j$ of its parameters so that it works well on all $K$ test cases.

Most of the training methods are based on the assumption that the real numbers $x_i^{(k)}$ that form the input (intensities, in case of voice recognition) are precisely known. In these methods, we "train" the model (i.e., adjust its parameters) until it precisely predicts the correct answer for all test inputs, i.e., until for every $k$ from 1 to $K$, we get the exact equality $f(x_1^{(k)}, \ldots, x_n^{(k)}, c_1, \ldots, c_p) = y^{(k)}$. In reality, since the test data are only approximately known, there is not much sense in achieving the exact fit. Moreover, if we do not stop after a "reasonable fit," in which the difference between the predicted values and the test data $y^{(k)}$ is of the same order as the test data inaccuracy, then not only we waste time on trying to adjust to error-caused "details," but we often worsen the fit by this adjustment: indeed, if, e.g., the actual dependency is linear, but we try to fit the noisy data exactly by a higher order polynomial, then, for large $x_i$, the higher order terms, that are caused by noise only, will grow much larger than the correct linear terms and make predicted values way off.

To avoid that, the authors of [8] designed an *interval*-based neural network, in which the test inputs and outputs are explicitly given as intervals $(\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_n^{(k)}, \mathbf{y}^{(k)})$, and adjustment is performed until for all input data, the prediction interval $f(\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_n^{(k)}, c_1, \ldots, c_p)$ becomes consistent with the output $\mathbf{y}^{(k)}$, i.e., until the corresponding intervals intersect.

For voice recognition, not only this method saves time, but it also leads to better predictions (by avoiding the "overlearning").

*Comment.* In addition to such novel ideas, other papers have been presented that describe the new sources of interval uncertainty or the use of this uncertainty in data processing and in control.

## 3. Additional source of interval uncertainty

Above, we have described two sources of interval uncertainty. In [3], a third source of interval uncertainty is analyzed that is caused by the following: in classical (two-valued) logic, every propositional formula, e.g., $A \rightarrow B$, can be described in terms of three basic operations: "and" ($\wedge$), "or" ($\vee$), and "not" ($\neg$). There are two basic ways of representing any propositional expression in this form: Disjunctive Normal Form (DNF)

$$(a \wedge b \wedge c) \vee (d \wedge \cdots) \vee \cdots$$

and Conjunctive Normal Form (CNF)

$$(a \vee b \vee c) \wedge (d \vee \cdots) \wedge \cdots$$

both forms are actively used in digital design. Because of this result, it is sufficient to implement the three basic operations; hence, many programming languages support only these operations, and standard hardware design blocks are usually also these three (plus their simplest combinations such as "nand" and "nor"). As a result, in expert systems, also analogues of these three operations are described. However, for multiple-valued logics used in expert systems, the three basic operations are no longer sufficient; CNF and DNF forms are, in general, different, and values resulting from different (classically equal) representations of the same propositional formula form an *interval* of possible values.

## 4. Applications of interval data processing: financial prediction, pattern recognition

In [12], a linear model with interval coefficients is used to predict the *prices of homes* and the *currency exchange rate*.

Another application, to *pattern recognition*, is presented in [2]. This application is based on the following idea. It is well known that a function is constant iff its derivative is equal to 0 (e.g., a position $x$ does not change iff the velocity $v$ is always zero). A natural discrete analogue of this result is: the sequence $x_i$ is constant iff all its first differences $v_i = x_i - x_{i-1}$ are equal to 0. In real life, measurements are never absolutely accurate. Therefore, a natural question appears: if the values $v_i = x_i - x_{i-1}$, $1 \leq i \leq n$ are close to 0, how close to a constant the values $x_i$ can be? In precise terms, what is the smallest value of $\Delta$ for which all the values $x_i$ are within $\Delta-$distance from some constant $c$ (i.e., for which there exists $c$ such that for all $i$ we have $x_i \in [c - \Delta, c + \Delta]$)?

If we know the values $x_i$ *exactly*, then we can estimate the desired "closeness" $\Delta$ as follows: From $v_i = x_i - x_{i-1}$, we can conclude that $x_1 = x_0 + v_1$, $x_2 = x_1 + v_2 = x_0 + (v_1 + v_2)$, and, in general, that $x_i = x_0 + s_i$, where $s_0 = 0$ and $s_i = v_1 + \cdots + v_i$ for $i > 0$. The largest of these values corresponds to the largest of $s_i$, and the smallest of them to the smallest of $s_i$. So, all the values $x_i$ lie in the interval $[x_0 + \min s_i, x_0 + \max s_i]$. It is easy to check that the smallest value of $\Delta$ is attained when we take as $c$, the midpoint of this interval; then $\Delta$ is equal to its half-width $\Delta = w/2$, where $w = \max\limits_i s_i - \min\limits_j s_j$. This width can be also expressed as $w = \max\limits_{i,j}(s_i - s_j)$, or, equivalently, as

$$w = \max_{i<j} \left| \sum_{p=i}^{j} x_p \right|.$$

This expression is a *norm* on the set of all sequences; under the name of a *discrepancy norm* it was first introduced, in essence, by H. Weyl in 1916 [13], and had several practical applications since then: e.g., H. Neunzert and B. Wetton, used discrepancy-norm techniques developed in their paper [9] in *quality control for fleece.*

In [2], this norm is applied to the following pattern recognition problem: how to automatically distinguish between a pixel in a (chaotic) picture, in a halftone (a more or less regular pattern of two or more color), at an edge (e.g., a caricature consisting of a few curves), and in a homogeneous area (one color only). Looking at the 8 nearest neighbors of a pixel, the variance is clearly high for a picture pixel, and (very) low for one in a homogeneous area. The non-trivial problem is to distinguish between edges and halftones (which are, visually, very different but may have the same variance). To solve this problem, the authors of [2] take into consideration that an edge pixel typically possesses a connected string of neighboring pixels for which the intensity $I_i$ is higher or lower than the average $\bar{I}$, which is highly unlikely for halftones. Thus, an edge pixel typically has much larger values of the discrepancy norm $w = \max\limits_{i<j} | \sum_{p=i}^{j} x_p |$, where $x_p = I_p - \bar{I}$. It turned out that, indeed, with the help of this norm it was possible to easily tell an edge from a halftone.

A similar problem occurs when we *do not know* the exact values of $v_i$ $(= x_i - x_{i-1})$. In particular, we may only know the value $\Delta$ for which, for some constant $c$, $x_i \in [c - \Delta, c + \Delta]$ for all $i$. In this situation, we may be interested in the interval of possible values of a linear functional $L = \sum c_i v_i$. If we substitute the expression for $v_i$ in terms of $x_i$, we get a formula

$$L = x_0(-c_1) + x_1(c_1 - c_2) + \cdots + x_{n-1}(c_{n-1} - c_n) + x_n(c_n).$$

The largest value of $L$ is attained when the values $x_j$ for which the coefficient is positive take the largest possible value $c + \Delta$, and the values $x_j$ for which the coefficient is negative take the smallest possible value $c - \Delta$. As a result, the largest possible value of $L$ is equal to

$$\Delta \cdot (|c_1| + |c_2 - c_1| + \cdots + |c_n - c_{n-1}| + |c_n|)$$

(and the smallest possible value of $L$ is minus this expression).

From the mathematical viewpoint, the variation-like expression in parenthesis is a *dual norm* to the discrepancy norm.

## 5. How to avoid interval overestimation? General idea and its application to interval probability estimates

If we know intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of possible values of input data $x_1, \ldots, x_n$, and if we know an algorithm $y = f(x_1, \ldots, x_n)$ that relates $x_i$ and $y$, then we can apply the following "naive interval computations" to estimate the interval $\mathbf{y}$ of possible values of $y$: on each step of the algorithm $f$, we replace the original arithmetic operation by the corresponding operation of interval arithmetic. It is well known that this method often leads to a drastic overestimation because it does not take into consideration that, first, the original values may have been related by a constraint, and second, that the intermediate values are not independent.

Traditionally, interval computation designs methods to deal with the second type of dependency, dependency of the intermediate data. However, in many real-life situations, the initial dependency is equally important. Such situations are analyzed, in the general case, in [6]. In [10], this general analysis is applied to the situation in which the initial data are the (intervally known) probabilities of different alternatives, constrained by a relation $x_1 + \cdots + x_n = 1$. For this case, explicit formulas are given, e.g., for the intervals of possible values of Bayes-updated probabilities $y$.

## 6. How to avoid overestimation? Specific idea: generalized intervals and their use in dynamical systems

If we are interested in the *static* quantity $x$ (that does not change with time), then the interval of possible values is, usually, the most adequate description of our knowledge. But in many real-life situations, the quantity $x$ that we are interested in is changing with time. Can we represent our knowledge about its values $x(t_1), \ldots, x(t_n)$ at different moments of time $t_1, \ldots, t_n$?

One possibility is to simply store the values (or intervals) that correspond to different moments of time. However, this takes lots of space. We can decrease the required storage space if we take into consideration that the dependency of $x(t)$ on time is usually *smooth*. Smooth means, in particular, than during a reasonably small time interval, the dependency $x(t)$ on $t$ can be well described by a linear function $x(t) \approx a + b(t - t_1)$. Hence, a reasonable way to compress the data is to store the coefficients $a$ and $b$ of this linear functions instead of the values $x(t_1), \ldots, x(t_n)$. The physical meaning of $a$ and $b$ is usually clear: e.g., if $x(t)$ is a coordinate of a particle at a certain point, then $a$ is its initial position, and $b$ is the particle's velocity. This physical interpretation can be used to determine $a$ and $b$ from the input data: e.g., we can take $a = x(t_1)$ and $b = (x(t_n) - x(t_1))/(t_n - t_1)$.

These formulas work perfectly well if inaccuracy is negligible: in this case, we can uniquely reconstruct $x(t)$ from $a$ and $b$. At first glance, a similar idea can be used if the inaccuracy is no longer negligible, i.e., if we do not know the actual values $x(t_i)$, but instead, know that these values belong to the known *intervals* $\mathbf{x}_i$. In this case, we have the interval $\mathbf{a} = \mathbf{x}_1$ of possible initial positions $x_1$, and we can use the above formula for $b$ to estimate the interval of possible values of velocity

$$\mathbf{b} = \frac{1}{t_n - t_1}\big(\mathbf{x}(t_n) - \mathbf{x}(t_1)\big).$$

The problem is that this compression *loses* information: if we reconstruct the interval for $x(t_n)$ as $\mathbf{a}+\mathbf{b}\cdot(t_n - t_1)$, we get a much wider interval than the original $\mathbf{x}_n$. One can easily see that this increase indeed happens on the simplest possible example of $n = 2$ and $\mathbf{x}(t_1) = \mathbf{x}(t_2) = [-1, 1]$.

To avoid this increase, the authors of [5] propose the following idea: we want to describe the dependency on the interval $\mathbf{x} = [\underline{x}, \overline{x}]$ on time. Mathematically, an interval is nothing else but two numbers. So, let us describe, separately, the dependency of $\underline{x}(t)$ on time, and the dependency of $\overline{x}$ on time, as, correspondingly, $\underline{x}(t) = \underline{a} + \underline{b} \cdot t$ and $\overline{x}(t) = \overline{a} + \overline{b} \cdot t$ for some $\underline{a}$, $\overline{a}$, $\underline{b}$, $\overline{b}$. If we formally combine these two descriptions, we get the representation $\mathbf{x}(t) = [\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] \cdot t$, which is very similar to what we tried before, except that the "interval" $[\underline{b}, \overline{b}]$ may now have a lower bound greater than the upper bound, i.e., it may be a *generalized* interval. In [5], such generalized intervals are successfully used to describe dynamical processes.

## 7. Multi-dimensional analogues of intervals: applications in pattern recognition

In one-dimensional case, intervals are a natural description of uncertainty. In the multi-dimensional case, the actual areas of uncertainty may be very complicated, so, it is necessary to use some approximating finite-parametric families of sets.

Different families have been proposed for this purpose: ellipsoids, boxes, more complicated polyhedra, etc. Each family has its advantages and drawbacks: e.g., boxes are easier to process computationally (e.g., intersection of two boxes is a box), but their boundaries are not smooth and therefore, they are less natural in approximating regions with smooth boundaries than, say, ellipsoids. Similarly, if we are solving a differential equation, for which we know the initial interval uncertainty, then boxes are easier to compute but the errors computed by using boxes grow faster than errors computed by using ellipsoids.

If we use boxes (as opposed to ellipsoids) to describe regions with smooth boundaries in real-life pattern recognition problems, then:

- on one hand, we need more boxes than ellipsoids to describe each region;
- but, on the other hand, we need fewer computation time to process each box than to process each ellipsoid.

So, if we choose, e.g., computation time until a certain approximation accuracy as a criterion for choosing a family of sets, then these two factors work in opposite directions, and it is difficult to predict which family of sets will be better.

The authors of [1] experimentally compared different families of sets on different real-life examples, including the iris data (*de facto* standard benchmark of pattern recognition problems) and biomedical data on blood cells. Surprisingly, in all real-life examples, ellipsoids turned out to be consistently better.

## 8. Multi-dimensional analogues of intervals: applications in control (in brief)

Another application of multi-dimensional analogues of intervals is to *control*. This is done in two papers that use different types of domains:

- Conic, spherical, planar, and other types of uncertainty are analyzed in [7];
- polyhedral uncertainty is analyzed in [4].

These two papers provide criteria under which dymanical systems are stable for all possible values of parameters from the given multi-dimensional domains.

## 9. City and conference

New Orleans is known as "the Big Easy," or "a city that care forgot." It is most famous for its annual Mardi Gras carnival, for its invention of jazz, for its unusual cuisine, and for its sleepless French Quarter, where jazz still rules. The city's last attempt at more "serious" fame was in 1815, when the future US President Andrew Jackson successfully defeated the British fleet. Ironically, it happened *after* the peace treaty was officially signed: communication was slow then.

Like many parts of Holland or St. Petersburg, several areas of New Orleans are built on the land conquered from the sea. Cemeteries are built above the land, and when you walk along one of the many below-sea-level streets, the site of ships passing by way above your head makes the world around feel very un-real.

Our banquet was at the Aquarium of the Americas, one of the world largest aquariums, where we could not only watch the exotic sea creatures, but also pat them (including a dangerous and teethy baby shark!), and... try how they taste.

## Acknowledgment

One of the authors (V. K.) was partly supported by the NSF Grant No. EEC-9322370. The authors are thankful to Ulrich Bodenhofer for the valuable comments.

## References

[1] Abe, S. and Thawonmans, R. *Fast training of a fuzzy classifier with ellipsoidal regions.* In: [11], Vol. 3, pp. 1875–1880.

[2] Bauer, P., Bodenhofer, U., and Klement, E. P. *A fuzzy algorithm for pixel classification based on the discrepancy norm.* In: [11], Vol. 3, pp. 2007–2012.

[3] Çiftçibaşi, T. and Altunay, D. *Fuzzy propositional logic and two-sided (intuitionistic) fuzzy propositions.* In: [11], Vol. 1, pp. 432–438.

[4] Filev, D. *Polytopic TSK fuzzy systems: analysis and synthesis.* In: [11], Vol. 1, pp. 687–693.

[5] Kandel, A., Friedman, M., and Ming, M. *On fuzzy dynamical processes.* In: [11], Vol. 3, pp. 1813–1818.

[6] Klir, G. J. and Cooper, J. A. *On constrained fuzzy arithmetic.* In: [11], Vol. 2, pp. 1285–1287.

[7] Marin, J. P. and Titli, A. *Robust performances of closed-loop fuzzy systems: a global Lyapunov approach.* In: [11], Vol. 1, pp. 732–737.

[8] Nava, P. and Traylor, J. M. *Speaker independent voice recognition with a fuzzy neural network.* In: [11], Vol. 3, pp. 2049–2052.

[9] Neunzert, H. and Wetton, B. *Pattern recognition using measure space metrics.* Techn. Rep. 28, Universität Kaiserslautern, Fachbereich Mathematik, 1987.

[10] Pan, Y., Klir, G. J., and Yuan, B. *Bayesian inference based on fuzzy probability.* In: [11], Vol. 3, pp. 1693–1699.

[11] *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems FUZZ–IEEE '96*, New Orleans, September 8–11, 1996, Vol. 1–3.

[12] Tanaka, H., Koyama, K., and Lee, H. *Interval regression analysis based on quadratic programming.* In: [11], Vol. 1, pp. 325–329.

[13] Weyl, H. *Über die Gleichverteilung von Zahlen Mod. Eins.* Math. Ann. 77 (1916), pp. 313–352.

[14] Zadeh, L. A. *The key roles of information granulation and fuzzy logic in human reasoning.* In: [11], Vol. 1, p. 1.

Erich Peter Klement
Fuzzy Logic Laboratorium Linz-Hagenberg
Institut für Mathematik
Johannes Kepler Universität
A-4040 Linz
Austria
E-mail: `klement@flll.uni-linz.ac.at`

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968
USA
E-mail: `vladik@cs.utep.edu`