9th Int. Workshop on IFSs, Banská Bystrica, 8 October 2013
Notes on Intuitionistic Fuzzy Sets
Vol. 19, 2013, No. 2, 57–70

An intuitionistic fuzzy estimation of the area of 2D–figures

Evgeni Marinov[1], Emilia Velizarova[2] and Krassimir Atanassov[1]

[1] Bioinformatics and Mathematical Modelling Department
Institute of Biophysics and Biomedical Engineering
Bulgarian Academy of Sciences
105 Acad. G. Bonchev Str., Sofia 1113, Bulgaria
e-mail: evgeni.marinov@gmail.com, krat@bas.bg

[2] Institute of Forest Research
Bulgarian Academy of Sciences
132, St. Kliment Ohridski Blvd, Sofia 1756, Bulgaria
e-mail: velizars@abv.bg

Abstract: An iterative procedure is proposed, starting from an initial grid-step and ending up with a smaller grid-step – small enough to be able to build the square hull for the given iteration. We propose in this paper also a formula for intuitionistic fuzzy estimation for the area surrounded by a continuous simple closed curve in the real 2D space. Therefore, this is a numerical method allowing to program the algorithm in any procedural language. The iterative process stops when a small enough limit between the upper and lower estimation has been reached.
Keywords: Square hull, Inner and outer polygon, Intuitionistic fuzzy estimation.
AMS Classification: 03E72, 51M25.

1 Problem statement

We propose in this paper a formula for intuitionistic fuzzy estimation for the area surrounded by a continuous simple closed curve in the real 2D space, i.e. the area of its interior. By a *simple* curve we mean one that has no self intersections and if, moreover, the curve is continuous that implies that its interior is a *simply connected* domain (cf. Munkres [4], Ch. 9). Given a 2D Cartesian coordinate system Oxy and a simple curve parametrized by

$$\vec{r}(t) = (r_1(t), r_2(t)) \colon [0,1] \longrightarrow \mathbb{R} \times \mathbb{R}, \tag{1}$$

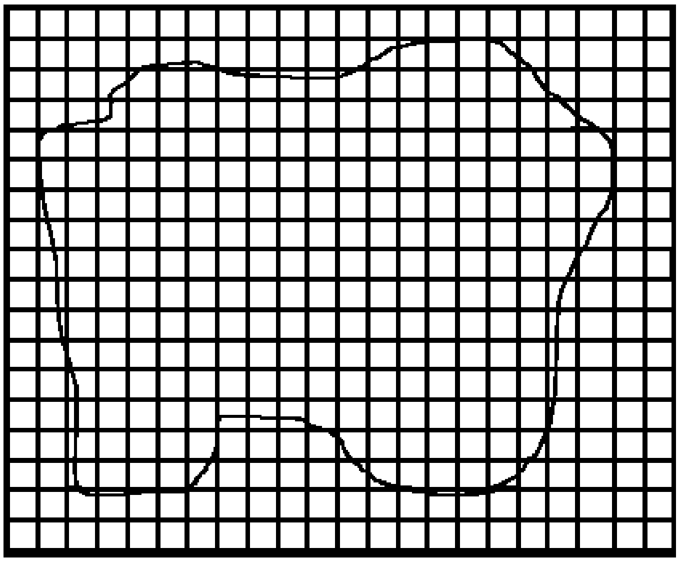

Figure 1: Simple curve and a grid with lines parallel to the two axes.

we are going to split the underlying space in a grid with lines parallel to the two axes (see Fig. 1.). This mesh has to be fine enough for a good estimation of the given curve. To be explained later what exactly "*fine enough grid*" means. We will, moreover, at the end consider only a smallest part of the grid with borders, consisting of lines parallel to the two axes, which enclose the curve. In order to define the intuitionistic fuzzy measure of the area enclosed by the initial curve we are going to introduce a procedure that provides the so called *square hull* and the two corresponding square polygons as well. The *inner* and *outer* square polygons corresponding to the square hull will be used for the actual intuitionistic fuzzy estimation. Therefore, the intuitionistic fuzzy estimation provides a variation of the real area enclosed by the curve $\vec{r}$.

2 Building the square hull

Taking initially the grid with a step of length l_0, supposed to be sufficiently small, we pass along the curve with $\vec{r}(t)$ letting the parameter t to vary from 0 to 1. Along its path, $\vec{r}(t)$ intersects the lines of the grid in different points and may pass through various squares of the grid leaving/entering them through their vertices or edges. Considering squares we mean by default the square taken with its boundary, i.e. together with the four corresponding edges and vertices in particular. Let us take all the squares of the grid to be colored white in the beginning. Then passing simulteniously through the squares on the path we color them in black.

Definition 1 *Let us define a semi-block neighbour of a square as the family of the square itself plus its three neighbouring squares such that they build a bigger square-block of four squares. In this way, we see that any square in an infinite grid has four semi-block neighbours. We will write: first, second, third and fourth semi-block corresponding to the four quadrants of the coordinate*

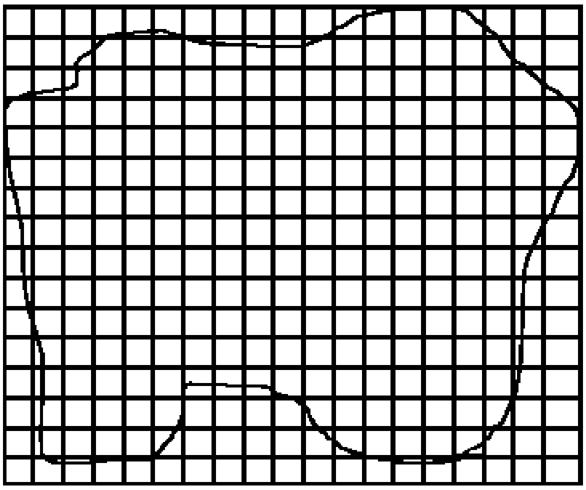

Figure 2: Minimal rectangular and with edges parallel to the axis part of the grid, containing the curve.

system in $\mathbb{R} \times \mathbb{R}$. Any two of the four semi-block neighbours of the given square can intersect in one or two squares where the initial square always belongs to their intersection.

Just after the square has been colored we check the color of its semi-block neighbours. If all the three squares belonging to any of the semi-block neighbours are black then we stop and re-mesh the space $\mathbb{R} \times \mathbb{R}$ taking a grid with step $\frac{l_0}{2}$. Then starting again until we get another semi-block neighbour colored in black, in which case the grid-step becomes $\frac{l_0}{2^2}$ and so on. At some point, say after n tries starting to count the iterations from 0, the grid will become so fine (with step $\frac{l_0}{2^n}$) that we won't get any semi-block neighbours of black squares, i.e. squares on the path of the curve. Let us now describe more detailed how we move from square to square. Starting from the initial square for some $t > t_0 \geq 0$, $\vec{r}(t)$ is going to leave its boundary. Supposing that the procedure, to be described inductively, has reached some square (to be called the *current square* or just $C-$square) on the curve's path. After passing successfully by the procedure the $C-$square is automatically put in the so called *procedure stack* or just $P-$*stack*. The last element of the $P-$stack, to be called *P-square*, the square preceding P-square in the stack is denoted by P' and the first (initial) and second square in the stack, P_0 and respectively P_1-*square*. After some square has became $C-$square and satisfies some conditions, to be described bellow, it is colored in black put in the $P-$stack and then the next square on the path becomes current. The procedure stops successfully if we reach the initial square (i.e. P_0) at some step. The grid is principally infinite but for the sake of simplicity we will concentrate ourselves on the minimal part of the grid, which contains the curve as shown on Fig. 2. This constraint is important because on the basis of this picture we are going to introduce intuitionistic fuzzy estimation in the next section. There are two main options and let us describe them with the whole substeps of the procedure.

- $\vec{r}(t)$ **leaves the P-square and enters the interior of some of its white neighbours called $C-$square.**
  That is, leaving the P-square $\vec{r}(t)$ does not move over some neighbour's edge (grid-line) or if it does then it has to enter its interior before leaving the edge. That is the easier case, where we first check the colors of all the seven neighbours of the newly entered $C-$square (all the $C-$neighbours are eight but we skip the one just left by the curve's path). Suppose that $\vec{r}$ enters $C-$square at point $t_0 \in [0, 1]$. If some of the $C-$neighbours are black we have few cases. Otherwise, if none of the $C-$square neighbours is black, except P and except P as a edge-neighbour and eventually P' as a diagonal-neighbour, we do not have different IF–THEN–ELSE cases. Remark that P'-square is allowed only to be a diagonal-neighbour of the current square.

  - IF all $C-$square neighbours are white (except P and eventually P' as a diagonal-neighbour), then color $C-$square in black and put it in the $P-$stack.
    Continue with the procedure.
  - ELSE–IF only one OR two of the $C-$square neighbours are black (except P-square and except P-square being an edge-neighbour and P'-square being a diagonal-square) and it is the initial P_0-square OR they are P_0 as an edge-neighbour and P_1 as diagonal-neighbour. $C-$square becomes black and is added to the stack and moreover:

    * IF P_0 is an edge-neighbour of $C-$square, we come to two other cases.
      · IF $\vec{r}([t_0, 1]) \subset (C \cup P_0)$ then
      Stop the procedure successfully.
      · ELSE $\vec{r}([t_0, 1])$ intersects the interior of some square other than C or P_0 then
      Restart the procedure with a twice shorter grid-step.

    * ELSE–IF P_0 is a diagonal-neighbour of $C-$square the situation becomes more complicated. Let us consider the semi-block neighbour of C containing P_0 and denote the other two squares of the semi-block by A_1 and A_2 (they are white). We come now to few other cases:
      · IF $\vec{r}([t_0, 1]) \subset (C \cup P_0)$ then $\vec{r}$ connects to its initial point without leaving the borders of C and P_0.
      Stop the procedure successfully.
      · ELSE–IF $\vec{r}([t_0, 1]) \subset (C \cup P_0 \cup A_1)$ then
      – IF A_1 does not have any black neighbours except C, P_0 as edge-neighbours and eventually P_1 as diagonal-neighbour and $\vec{r}$ passes first through the interior of A_1 before entering the interior of P_0, then
      Continue with the procedure.
      – ELSE
      Restart the procedure with a twice shorter grid-step.

· ELSE–IF $\vec{r}([t_0, 1]) \subset (C \cup P_0 \cup A_2)$ then
– IF A_2 does not have any black neighbours except C, P_0 edge-neighbours and eventually P_1 as diagonal-neighbour and $\vec{r}$ passes first through the interior of A_2 before entering the interior of P_0, then
Continue with the procedure.
– ELSE
Restart the procedure with a twice shorter grid-step.

∗ ELSE
Restart the procedure with a twice shorter grid-step.

- **$\vec{r}(t)$ leaves the P-square through a vertex v_0 continuing on straight line over series of edges belonging to white squares only.**
Moving that way, we have then a few possibilities:

  – $\vec{r}(t)$ reaches a vertex v_1 belonging to black square(s). Let us build two sequences of squares to be useful in what follows. T_0 contains all squares on the same level (line/column) of P starting from the first square after P to the first square that contains v_1. T_1 contains all squares on the neighbour level (line/column) of P starting from the corresponding diagonal-neighbour of P to the first square that contains v_1. Because the path of edges (the straight line) from v_0 to v_1 contains only vertices which do not belong to any black squares the choice of T_0 and T_1 is possible. We say that T_i $(i = 1, 2)$ is a correct sequence of squares if it does not contain any vertices that belong to black squares except the first and the last elements of T_i.
We come across a few cases:

    ∗ IF none of the black squares containing v_1 is the initial square P_0 then
Restart the procedure with a twice shorter grid-step.

    ∗ ELSE–IF $v_1 \in P_0$, and P_0 and P lie on the same level.
In this case P_0 and P are bold-connected through T_0.

      · IF T_0 is a correct sequence of squares and $\vec{r}$ does not leave P_0 once entered it, then color the squares from T_0 black and put them in the $P-$stack.
Stop the procedure successfully.

      · ELSE–IF T_0 is not correct and T_1 is correct and $\vec{r}$ does not leave P_0 once entered it, then color the squares from T_1 black and put them in the $P-$stack.
Stop the procedure successfully.

      · ELSE
Restart the procedure with a twice shorter grid-step.

    ∗ ELSE–IF $v_1 \in P_0$, and P_0 and P lie on neighbour levels, then

      · IF T_1 is a correct sequence of squares and $\vec{r}$ does not leave P_0 once entered it, then color the squares from T_1 black and put them in the $P-$stack.
Stop the procedure successfully.

- ELSE–IF T_1 is not correct and T_0 is correct and $\vec{r}$ does not leave P_0 once entered it, then color the squares from T_0 black and put them in the $P-$stack. **Stop the procedure successfully**.
- ELSE
  Restart the procedure with a twice shorter grid-step.

– $\vec{r}(t)$ enters the interior of a white square, called now the current square C. Let us also denote v_0 the vertex of P which lies ot the straight line of edges belonging to $\vec{r}$ and c_0 - the vertex of C closest to v_0. By analogy as before, let us introduce $T_i (i = 0, 1)$, where T_0 lies on the same level of P and T_1 on the corresponding neighbour level. If C is on the same level as P then it is an edge-neighbour of the last element of T_0 (if not empty) and diagonal-neighbour of the last element of T_1 (if not empty). Otherwise, if C is on the corresponding neighbour level of P, then it is an edge-neighbour of the last element of T_1 (if not empty) and diagonal-neighbour of the last element of T_1 (if not empty).
We have now a few cases:

* IF C and P lie on the same level. We have then another two options.
  - IF T_0 is a correct sequence of squares,
    then color the squares from T_0 black and put them in the $P-$stack. The next iteration C will be the current square and handled appropriately.
    Continue with the procedure.
  - ELSE–IF T_0 is not correct and T_1 is correct,
    then color the squares from T_1 black and put them in the $P-$stack. The next iteration C will be the current square and handled appropriately.
    Continue with the procedure.
  - ELSE
    Restart the procedure with a twice shorter grid-step.
* IF C and P lie on the neighbour levels. We have then other two options.
  - IF T_1 is a correct sequence of squares,
    then color the squares from T_1 black and put them in the $P-$stack. The next iteration C will be the current square and handled appropriately.
    Continue with the procedure.
  - ELSE–IF T_1 is not correct and T_0 is correct,
    then color the squares from T_0 black and put them in the $P-$stack. The next iteration C will be the current square and handled appropriately.
    Continue with the procedure.
  - ELSE
    Restart the procedure with a twice shorter grid-step.

The procedure producing the square hull has been described above step by step.

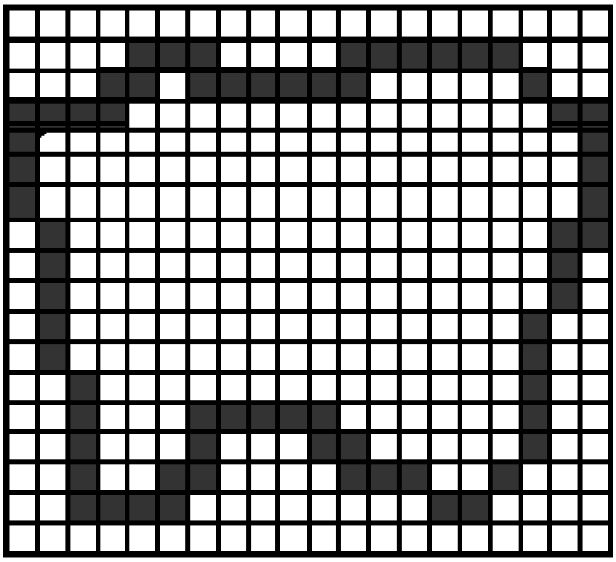

Figure 3: The square hull of the simple curve $\vec{r}$ at the end of the last step of the procedure.

Definition 2 (Square-hull) *After some number of iterations for some short enough grid-step the procedure stops successfully and we get a sequence (closed bold "curve") of squares and/or lines which contains the initial curve in consideration $\vec{r}$. Let us call it the square hull of $\vec{r}$ (see Fig. 3.).*

Some useful objects for the calculation of the intuitionistic fuzzy estimation of the area enclosed by $\vec{r}$ will be introduced through the following definition.

Definition 3 (Square-polygons) *The square hull provides us two polygons with edges parallel to the axis (Fig. 4.). Let us call them the inner square polygon $\mathcal{P}^i(\vec{r}, l)$ (see Fig. 5.), i.e. the one in the interior of $\vec{r}$ and the outer square polygon $\mathcal{P}^o(\vec{r}, l)$ (see Fig. 5.), i.e. the one laying in the infinite component of the complement of $\vec{r}$ - the outside of $\vec{r}$. Here by l is denoted the grid step at which the algorithm stops successfully. The area enclosed by $\mathcal{P}^i(\vec{r}, l)$ and $\mathcal{P}^o(\vec{r}, l)$ will be denoted by $\mathcal{A}^i(\vec{r}, l)$ and $\mathcal{A}^o(\vec{r}, l)$, respectively.*

If it is not ambiguously the polygons defined above will be called just outer and respectively inner polygons of $\vec{r}$. The introduced procedure provides the square hull in such a way that the inner square polygon is not empty and has no holes, which is its most important property. We omit here the mathematical proof of this but during the procedure when we come to such a case (where holes in the inner square hull arise) we intentionally stop the procedure and restart it with a shorter grid-step. We have used twice shorter grid-step than the current one, for instance. That is, initially the algorithm starts with an appropriately chosen grid-step of length l_0 and if it turns out to be to large for our purposes, using a decreasing function

$$g\colon (0, l_0] \longrightarrow (0, l_0) \tag{2}$$

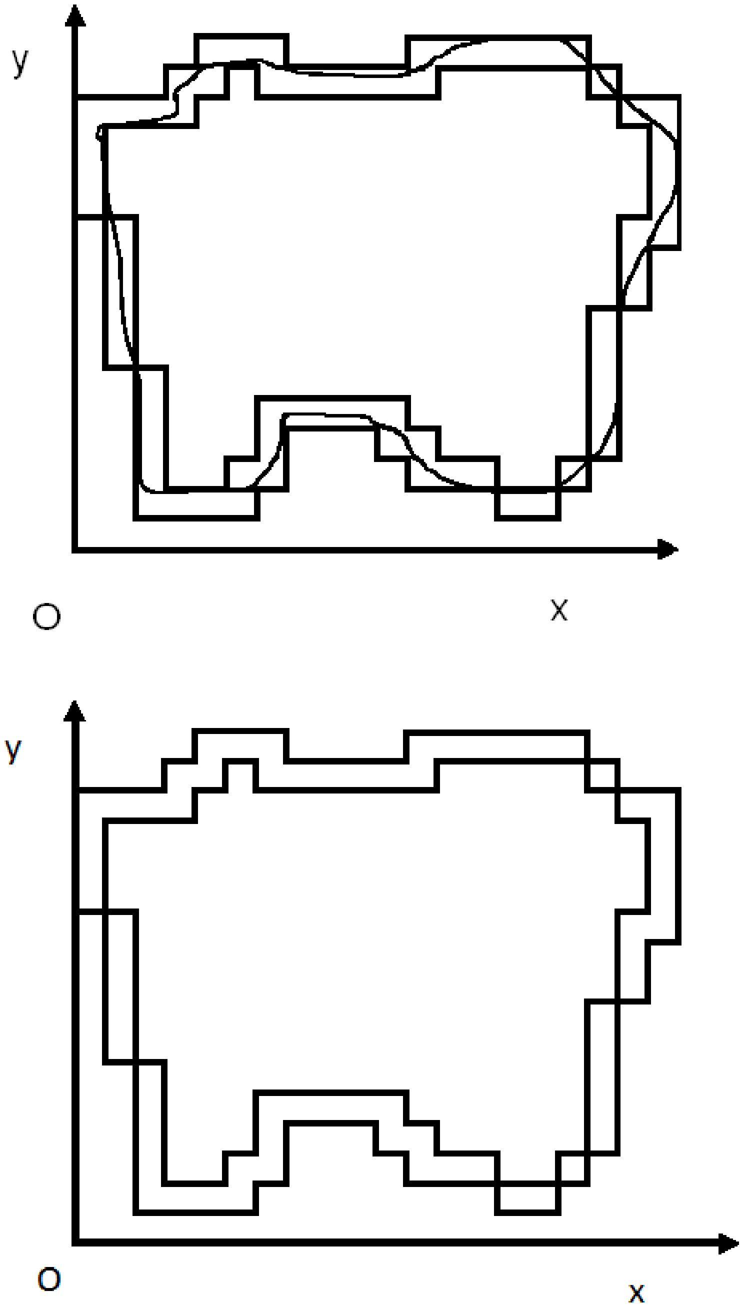

Figure 4: The inner and outer square polygons at the end of the last step of the procedure.

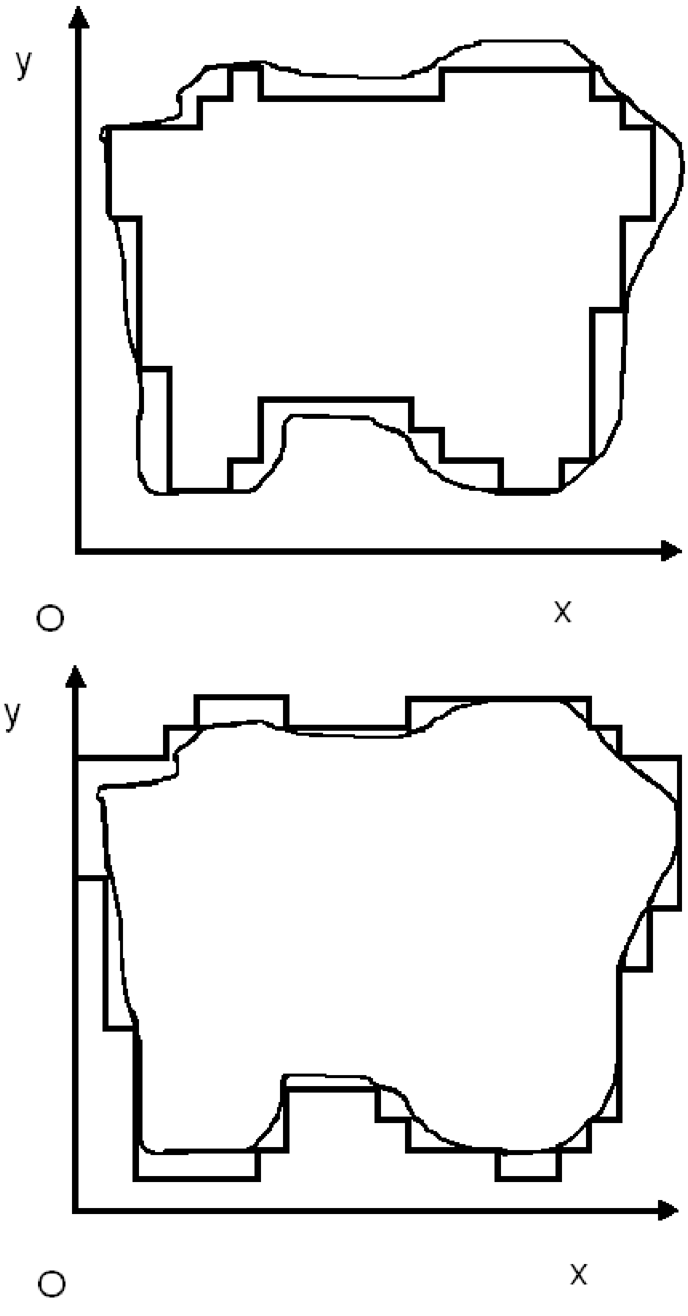

Figure 5: The inner and outer polygon approximation of the simple curve $\vec{r}$ at the end of the last step of the procedure.

we progressively shorten the grid-step with each iteration. Starting with 0 at the nth iteration take a grid-step length equal to $g^n(l_0)$. Because the function g is decreasing we have that: $l_0 > g(l_0) > g^2(l_0) > \ldots > g^n(l_0) > 0$ for all $n \in \mathbb{N}$. In the proposed procedure we actually used a grid step function: $g(t) = \frac{t}{2}$.

At the end of the procedure we also have the $P-$stack containing series of the squares through which the curve passes. We can do a loop over all these elements, already having their coordinates, i.e. the coordinates of the nodes of each square. Iterating all the squares we compute and hold the upper node and the most lower node, which means the node from the $P-$stack having the largest $y-$coordinate and resp. the node having the lowest $y-$coordinate. We compute the largest and lowest $x-$coordinates for nodes belonging to squares from $P-$stack. Let us denote by $Grid(l)$ the integer grid points corresponding to step l, i.e. $(x_0, y_0) \in Grid(l)$ if and only if $x_0, y_0 \in l\mathbb{N}$. We consider here only the first quadrant, where all points (x_0, y_0) have positive coordinates. With no loss of generality, we can assume that the figure in consideration lies there with its approximation, i.e. the corresponding square hull. For a node with coordinates (x, y) we write $(x, y) \in P-$stack if (x, y) is a node belonging to any square element of the $P-$stack. Supposing that the procedure ends with step l, let us now denote,

$$Y_{max} = \max\{y \mid (x, y) \in P - stack\}, Y_{min} = \min\{y \mid (x, y) \in P - stack\}$$

And the corresponding definitions on the $x-$axis,

$$X_{max} = \max\{x \mid (x, y) \in P - stack\}, X_{min} = \min\{x \mid (x, y) \in P - stack\}$$

We can now cut from the whole grid only the big square with nodes

$$(X_{min}, Y_{min}), (X_{max}, Y_{min}), (X_{min}, Y_{max}), (X_{max}, Y_{max})$$

Let us take a new coordinate system derived from the original one translated to (X_{min}, Y_{min}) and concentrating ourselves only on the rectangle described above (see Fig. 6.). This rectangle has a surface with measure to be denoted $\mathcal{S}(\vec{r}, g, l_0)$. As an example on Fig. 6. is plotted the minimal rectangular area of the initial simple curve $\vec{r}$ given with Fig. 1., where $\mathcal{S}(\vec{r}, g, l_0) = 20 \times 16 \times l_1 = 320 \times l_1$ and l_1 is the final grid step after the work of the procedure.

This provides a more convenient picture for practical manipulation of the result and for the calculation of intuitionistic fuzzy estimation to be defined in the next section.

The above definitions and observations are exactly what we need for the intuitionistic fuzzy estimation of the area enclosed by $\vec{r}$. Let us first give a brief introduction to the intuitionistic fuzzy sets and state the IFS estimation formula in terms of membership and non-membership functions.

3 Intuitionistic fuzzy estimation of the inner area

The notion of intuitionistic fuzzy set (or abbreviated as IFS) provides a very intuitive and natural tool for an adequate estimation of the area enclosed by a simple continuous curve. As an application of this method we give an estimation for the area of a forest fire spread.

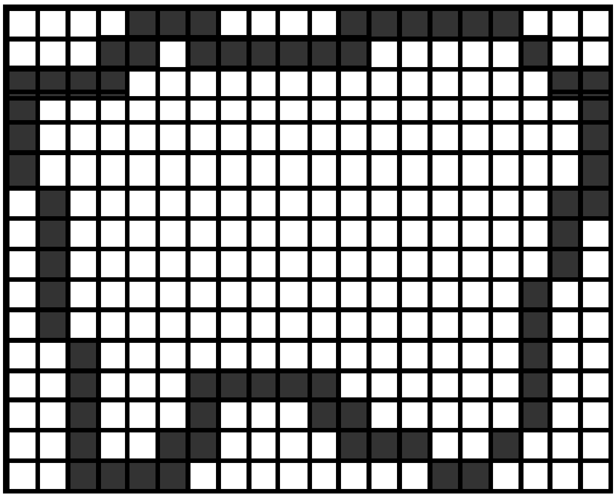

Figure 6: The minimal rectangular area from the grid that contains the simple curve $\vec{r}$. Its surface $\mathcal{S}(\vec{r}, g, l_0)$ $\vec{r}$ is equal to $20 \times 16 \times l_1$, where l_1 is the final grid step after the work of the procedure.

A fuzzy set in X (cf. Zadeh [7]) is given by

$$A^{'} = \{\langle x, \mu_{A'}(x)\rangle | x \in X\} \tag{3}$$

where $\mu_{A'}(x) \in [0, 1]$ is the *membership function* of the fuzzy set $A^{'}$. As opposed to the Zadeh's fuzzy set (abbreviated FS), Atanassov extended its definition to an intuitionistic fuzzy set (IFS) (cf. [1] and [2]) A, given by

$$A = \{\langle x, \mu_A(x), \nu_A(x)\rangle | x \in X\} \tag{4}$$

where: $\mu_A : X \to [0, 1]$ and $\nu_A : X \to [0, 1]$ such that

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1 \tag{5}$$

and $\mu_A(x)$, $\nu_A(x) \in [0, 1]$ denote a *degree of membership* and a *degree of non-membership* of $x \in A$, respectively. An additional concept for each IFS in X, that is an obvious result of (4) and (5), is called *hesitation margin* of $x \in A$ and defined by:

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x). \tag{6}$$

It expresses the lack of knowledge of whether x belongs to A or not (cf. [1]). It is obvious that $0 \leq \pi_A(x) \leq 1$, for each $x \in X$.

Hesitation margins turn out to be relevant for both - applications and the development of theory of IFSs. For instance, distances between IFSs are calculated in the literature in two ways, using two parameters only (see e.g. [1]) or all three parameters (cf. [3], [5], [6]). Both ways are proper from the point of view of pure mathematical conditions concerning distances but one cannot say that both ways are equal when assessing the results obtained by the two approaches.

As described in the previous section the iterative procedure starts with inputs – the simple continuous curve $\vec{r}(t)$, an initial sufficiently small grid-step l_0. We have also introduced a decreasing step function g which can be used in the algorithm where there it is taken to be $g(t) = \frac{t}{2}$. The algorithm ends up at an iteration, say $j(\vec{r}, g, l_0) = j_0$. And therefore, the end grid step becomes $l_1 = g^{j_0}(l_0)$. In what follows, we denote the already defined $\mathcal{S}(\vec{r}, g, l_0)$ by $\mathcal{S}_0$ for fixed curves, step function and initial step. Let us give, on the basis of the already defined $\mathcal{A}^o$ and $\mathcal{A}^i$, an intuitionistic fuzzy estimation.

Definition 4 *In the above notations let us define*

$$\mu_{(\vec{r},g,l_0)}(1) = \frac{\mathcal{A}^i(\vec{r}, l_1)}{\mathcal{S}_0} \text{ and } \nu_{(\vec{r},g,l_0)}(1) = \frac{\mathcal{S}_0 - \mathcal{A}^o(\vec{r}, l_1)}{\mathcal{S}_0}$$

with $l_1 = g^{j(\vec{r},g,l_0)}(l_0)$, which obviously take values in $[0, 1]$. In a similar way, repeating the main procedure with the same input parameters $\vec{r}, g$ and initial grid step $l_1^{'} = g(l_1) = g^{j(\vec{r},g,l_0)+1}(l_0)$, we define

$$\mu_{(\vec{r},g,l_0)}(2) = \frac{\mathcal{A}^i(\vec{r}, l_2)}{\mathcal{S}_0} \text{ and } \nu_{(\vec{r},g,l_0)}(2) = \frac{\mathcal{S}_0 - \mathcal{A}^o(\vec{r}, l_2)}{\mathcal{S}_0}$$

More generally, let us inductively define $l_k = g^{j(\vec{r},g,l_{k-1}^{'})}(l_{k-1}^{'})$, for any positive integer k and

$$\mu_{(\vec{r},g,l_0)}(k) = \frac{\mathcal{A}^i(\vec{r}, l_k)}{\mathcal{S}_0} \text{ and } \nu_{(\vec{r},g,l_0)}(k) = \frac{\mathcal{S}_0 - \mathcal{A}^o(\vec{r}, l_k)}{\mathcal{S}_0}$$

where for initial grid-step of the k-th start of the procedure we have used $l_{k-1}^{'} = g(l_{k-1})$.

Taking in consideration the last definition, we may write down the hesitation margin for the k-th step as $\pi_{(\vec{r},g,l_0)}(k) = 1 - \mu_{(\vec{r},g,l_0)}(k) - \nu_{(\vec{r},g,l_0)}(k)$. Therefore, we have that

$$\pi_{(\vec{r},g,l_0)}(k) = \frac{\mathcal{A}^o(\vec{r}, l_k) - \mathcal{A}^i(\vec{r}, l_k)}{\mathcal{S}_0}$$

which is exactly the intuitionistic fuzzy estimation of the hesitation margin – the normalized area measure of the corresponding square-hull. It is obvious that for $k_1 < k_2$ we have that $\pi_{(\vec{r},g,l_0)}(k_1) > \pi_{(\vec{r},g,l_0)}(k_2)$, which means exactly that $\pi_{(\vec{r},g,l_0)}$ is a decreasing function on the set of positive integer numbers $\mathbb{N}$. Therefore, we may suppose that we are given a small enough positive real number ϵ_0 based on the curve $\vec{r}$ which area has to be estimated. By means of the described algorithm we are computing then the iterative estimations (square-hulls) until a positive integer k has been reached for which $\pi_{(\vec{r},g,l_0)}(k) \leq \epsilon_0$. The last square-hull corresponding to the iteration number k then provides a satisfactory intuitionistic fuzzy estimation of the curve. This also means that $\mathcal{A}^o(\vec{r}, l_k)$ and $\mathcal{A}^i(\vec{r}, l_k)$ provide a corresponding satisfactory estimation of the area surrounded by the curve.

Example 1 *An example of a curve $\vec{r}$ has been provided through the pictures of this paper, see Fig. 6. For the corresponding areas we have, say after the end of the procedure started for first time with a grid-step l_0 and finishing with end grid-step l_1, that*

- $\mathcal{S}(\vec{r}, g, l_0) = \mathcal{S}_0 = 320 \times l_1$
- $\mathcal{A}^i(\vec{r}, l_1) = 187 \times l_1$
- $\mathcal{A}^o(\vec{r}, l_1) = 255 \times l_1$

Therefore, for the intuitionistic fuzzy estimations after the first application of the procedure, we have that

- $\mu_{(\vec{r},g,l_0)}(1) = \dfrac{187}{320}$
- $\nu_{(\vec{r},g,l_0)}(1) = \dfrac{320 - 255}{320} = \dfrac{65}{320}$
- $\pi_{(\vec{r},g,l_0)}(1) = \dfrac{255 - 187}{320} = \dfrac{68}{320}$

Let us remark that for the second, third, etc., time repeating the initial procedure we use the same $\mathcal{S}_0$, which is the output from the first start of the procedure with grid-step l_0.

4 Applications

We see now that the described procedure and the intuitionistic fuzzy estimation give an iterative numerical algorithm which can be implemented in any procedure programming language.

The method described in this paper can be adequately applied for an estimation of areas affected by forest and field fire spreads.

Acknowledgements

This work has been supported by the Bulgarian Science Fund under Grant DFNI-I-01-0006.

References

[1] Atanassov, K. *Intuitionistic Fuzzy Sets: Theory and Applications*. Springer-Verlag, Heidelberg, 1999.

[2] Atanassov, K. *On Intuitionistic Fuzzy Sets Theory*, Springer-Verlag, Berlin, 2012.

[3] Atanassov, K., V. Tasseva, E. Szmidt, J. Kacprzyk. On the geometrical interpretations of the intuitionistic fuzzy sets. *Issues in the Representation and Processing of Uncertain and Imprecise Information. Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets, and Related Topics* (Eds. Atanassov K., J. Kacprzyk, M. Krawczak, E. Szmidt), EXIT, Warsaw, 2005.

[4] Munkres, J. *Topology*, 2nd ed., Prentice Hall, 2000.

[5] Szmidt, E., J. Baldwin. New similarity measure for intuitionistic fuzzy set theory and mass assignment theory. *Notes on Intuitionistic Fuzzy Sets*, Vol. 9, 2003, No. 3, 60–76.

[6] Szmidt, E., J. Baldwin. Distances between intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, Vol. 114, 2000, No. 3, 505–518.

[7] Zadeh L. A. Fuzzy sets. *Information and Control*, Vol. 8, 1965, No. 3, 338–353.