Levels of Degeneracy and Exact Lower Complexity Bounds for Geometric Algorithms

Peter Hertling *   Klaus Weihrauch †

Abstract

Degenerate configurations cause great problems in the implementation of geometric algorithms. This paper provides a mathematical framework for the investigation of degeneracies, based on the observation that degeneracy is discontinuity. It is shown that one can define different levels of discontinuity and that the level of a computational problem gives a lower bound for the number of tests needed in a computation tree solving that problem. In many cases this lower bound is exact. We illustrate the concept of level by various simple examples from computational geometry. We show that for most of the geometric problems the level is in $O(n)$. Finally we discuss the computational problems caused by degeneracies.

1 Introduction

Usually a problem in computational geometry can be formalized by a partial function $f :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$ (we may assume $\mathbb{N} \subseteq \mathbb{R}$) or by a set of such functions.

The standard models for formulating geometric algorithms are the real random access machine (real RAM) and the decision tree ([13]). It is well known that serious difficulties may occur when geometric algorithms are implemented (see e.g. [8]). One reason for this are degeneracies and the fact that infinite precision arithmetic is not available on real world computers. A degenerate configuration $x \in \mathbb{R}^M$ can be informally decribed as a point where arbitrarily small movements of x may result in desastrous movements of $f(x)$. Mathematically this means discontinuity, therefore we formulate:

Thesis 1 *A configuration $x \in \mathbb{R}^M$ of a geometric problem $f :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$ is* degenerate, *iff f is* discontinuous *in x.*

In this contribution we introduce as a simple but presumably new concept the *level of discontinuity* for measuring the complicatedness of a degeneracy. The main goal of this paper is to study the relation between the level of discontinuity and the real RAM computational complexity. In the last section we discuss implementation problems.

As a computation model we introduce continuous computation trees (CCT's), which generalize the loop-free real RAM's. For a function f we define the *level of discontinuity of f at $x \in dom(f)$* and the *level* of f by analytic properties. For a CCT T we define the *size* of T at an input x and the *size* of T, which is the number of leaves of T. The size of T at x and the size of T can be interpreted as complexities. Yap ([21]) distinguishes between *problem-dependent* and *algorithm-dependent* degeneracies. In our context the level corresponds to problem-dependent degeneracies and the size corresponds to algorithm-dependent degeneracies.

Our first result states that for any CCT T and input x the size of T at x is at least as great as the level of f_T (the function computed by T) at x. This means that the level of f at x is a lower bound for the complexity at x for every CCT T computing f.

Our second result states that any function f of finite level n can be computed by a CCT T of optimal size, i.e. of size n. If $range(f)$ is discrete, then there is even a locally optimal CCT. Thus, the level of f is an exact lower complexity bound for the CCT's computing f. In fact, in many applications the lower bound can be achieved even with finite computation trees which contain only polynomial or rational functions.

We illustrate the results by various sorting problems and the line segment intersection problem

*Fernuniversität, D - 58084 Hagen, Germany. email: peter.hertling@fernuni-hagen.de

†Fernuniversität, D - 58084 Hagen, Germany. email: klaus.weihrauch@fernuni-hagen.de

from computational geometry ([14]) in Section 4. We prove that the level of a function $f :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$ with *analytic domains of continuity* is bounded by $M+1$. This applies to most of the geometric problems.

In the last section we discuss the computational problems caused by degeneracies.

2 Levels of degeneracy and size of CCT's

In this section we introduce the basic definitions and formulate the main theorems. In the following by $f :\subseteq X \longrightarrow Y$ we denote a partial function f with $dom(f) \subseteq X$ and $range(f) \subseteq Y$.

Definition 2.1 Consider $f :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$. Define sets A_n^f ($n \in \mathbb{N}$) inductively by $A_0^f := dom(f)$, $A_{n+1}^f := \{x \in A_n^f \mid f|_{A_n^f}$ is discontinuous at $x\}$ for all $n \geq 0$. For $x \in \mathbb{R}^M$ define the *level (of discontinuity) of f at x* by $lev(f,x) := \min\{n \mid x \notin A_n^f\}$ and define the *level (of discontinuity) of f* by $Lev(f) := \min\{n \mid A_n^f = \emptyset\}$.

(Set $\min(\emptyset) := \infty$.) Obviously $Lev(f) = \max\{lev(f,x) \mid x \in \mathbb{R}^M\}$. If $dom(f) \neq \emptyset$ and f is continuous then $Lev(f) = 1$. If f is continuous in x then $lev(f,x) = 1$. For example the simple test

$$f_1 : \mathbb{R} \longrightarrow \{0,1\}, \quad f_1(x) := \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

has level 2 ($A_0^{f_1} = \mathbb{R}$, $A_1^{f_1} = \{0\}$, $A_2^{f_1} = \emptyset$). If $lev(f,x) < lev(f,x')$ then intuitively f is more discontinuous in x' than in x. Thus the level indeed measures the discontinuity of f. Finer hierarchies of discontinuous functions are studied in [7].

In computational geometry algorithms are formulated in the real random access machine (real RAM) model ([13], [3]). Most of the problems from algorithmic geometry can be implemented by loop–free real RAM's. We generalize the loop–free real RAM's by admitting assignments $x := f(y)$ with arbitrary continuous functions $f :\subseteq \mathbb{R}^m \longrightarrow \mathbb{R}^n$. We define a normal form of such programs.

Definition 2.2 A CCT (*continuous computation tree*) T is a loop–free flowchart, where each internal node l is a binary branching with a test $f_l(x) < 0$ ($f_l : \mathbb{R}^M \longrightarrow \mathbb{R}$ continuous) and each leaf l contains an assignment of the form $y := g_l(x)$ ($g_l :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$ continuous), where x is the input and y is the result of the computation. By $f_T :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$ we denote the function computed by T.

Every loop–free real RAM with n branchings can be transformed into an equivalent CCT with at most n branchings. We define the size of T at input x and the size of T.

Definition 2.3 A test $g(x) < 0$ is called *critical* in a point x iff $x \in \overline{U} \setminus U$ where $U := \{x \mid g(x) < 0\}$. Let T be CCT and $x \in dom(f_T)$.

(1) Define $size(T,x)$ inductively by:
$size(T,x) := 1$ if T has no branchings.
If $T = (if\ g(x) < 0\ then\ T_1\ else\ T_2)$, then

$$size(T,x) := \begin{cases} size(T_1,x) + size(T_2,x) \\ \quad \text{if the test } g(x) < 0 \\ \quad \text{is critical in } x \\ size(T_1,x) \quad if\ g(x) < 0 \\ size(T_2,x) \quad else\,. \end{cases}$$

(2) Define
$Size(T) := 1 +$ number of branchings in T (= number of leaves in T).

Informally, $size(T,x)$ is the number of leaves in T which can be reached on input x, if in each step of the computation x may be disturbed independently. The *size* function, therefore, formalizes the idea of "level of degeneracy" of an algorithm. Obviously the total number of test labels in T which are critical in x is an upper bound for $size(T,x) - 1$.

Now, we formulate our main results.

Theorem 1 *Let T be a CCT and let $x \in dom(f_T)$. Then $lev(f_T,x) \leq size(T,x)$ and $Lev(f_T) \leq Size(T)$.*

Thus, $lev(f,x) - 1$ is an absolute lower bound for the number of tests in any CCT T which computes f in a neighbourhood of x. This holds especially for the loop–free real RAM's. By the following remarkable theorem the lower bounds from Theorem 1 are exact (in the case $lev(f,x)$ if $range(f)$ is discrete).

Theorem 2 *Let f be a function of finite level ≥ 1.*

a) There is a CCT T with $f = f_T$ and $Lev(f) = Size(T)$.

b) If $range(f)$ is discrete, then there is a CCT T with $f = f_T$ and $(\forall x \in dom(f))\ lev(f,x) = size(T,x)$.

The theorem also guarantees the existence of an optimal CCT T for any discontinuous function of finite level. If $range(f)$ is discrete, then there is even a locally optimal CCT. Very often there is already a loop–free real RAM with the minimal number of branchings (see the examples in Section 4).

3 Proof of Theorems 1 and 2

The following lemma contains some simple facts about the functions lev and $size$. The proof is left to the reader.

Lemma 3.1 *a) If two functions f and g coincide in a neighbourhood of a point x, then $lev(f,x) = lev(g,x)$.*

b) If a function f is a restriction of a function g, i.e. $f = g|_X$ for some set X, then $lev(f,x) \leq lev(g,x)$ for all points x.

c) If $x \in A_n^f$, then $lev(f,x) = n + lev(f|_{A_n^f}, x)$.

d) For any CCT T the function $size(T,.)$ is upper semicontinuous, i.e. for any point x there is a neighbourhood U of x such that $size(T,y) \leq size(T,x)$ for all $y \in U$.

Proof of Theorem 1

Let T be a CCT. The second inequality is an immediate consequence of the first because $Lev(f_T) = \max\{lev(f_T,x) \mid x \in dom(f_T)\}$ and $(\forall x \in dom(f_T))$ $size(T,x) \leq Size(T)$.

We prove the first inequality. If T does not contain a branching node, then f_T is a continuous function and we have for all $x \in dom(f_T)$:

$$lev(f_T,x) = 1 = size(T,x).$$

Else T has the form $T = (if\ \ g(x) < 0$ then T_1 else $T_2)$ where g is a continuous function and T_1 and T_2 are CCT's. Set $o := \{x \mid g(x) < 0\}$. The proof is done by induction.

If $x \in o$, then

$$\begin{aligned} lev(f_T,x) &= lev(f_{T_1},x) && \text{(by Lemma 3.1 a))} \\ &\leq size(T_1,x) && \text{(by induction)} \\ &\leq size(T,x)\,. \end{aligned}$$

The assertion follows in the same way if x is in the interior of the complement of o.

Now fix an $x \in \bar{o} \setminus o$. Then

$$size(T,x) = size(T_1,x) + size(T_2,x). \qquad (1)$$

By Lemma 3.1 d) there is an open neighbourhood U of x such that we have $size(T_1,y) \leq size(T_1,x)$ for all $y \in U$. For all $y \in U \cap o$ we obtain

$$lev(f_T,y) = lev(f_{T_1},y) \leq size(T_1,y) \leq size(T_1,x)$$

as above by Lemma 3.1 a) and inductive hypothesis. With $k := size(T_1,x)$ we conclude

$$A_k^{f_T} \cap U \subseteq o^c\,. \qquad (2)$$

If $x \notin A_k^{f_T}$, then $lev(f_T,x) \leq k = size(T_1,x) \leq size(T,x)$ finishes the proof. If $x \in A_k^{f_T}$, then by Lemma 3.1 c) and a)

$$lev(f_T,x) = k + lev(f_T|_{A_k^{f_T} \cap U}, x).$$

The relation (2) and $f_T|_{o^c} = f_{T_2}|_{o^c}$ imply

$$lev(f_T,x) = k + lev(f_{T_2}|_{A_k^{f_T} \cap U}, x)\,.$$

By Lemma 3.1 b) and the inductive hypothesis we obtain

$$lev(f_T,x) \leq k + size(T_2,x)\,.$$

Finally Equation (1) gives the assertion $lev(f_T,x) \leq size(T,x)$. □

In the rest of this section we always write A_n instead of A_n^f. For the proof of Theorem 2 we need a lemma, which we state without proof.

Lemma 3.2 *Let $f :\subseteq I\!R^M \longrightarrow I\!R^N$ be a function and n be an integer. Fix a point $x_0 \in \overline{A_n}$. Then $f|_{\overline{A_n}}$ is discontinuous in x_0 iff $x_0 \in A_{n+1}$.*

Proof of Theorem 2

a) If the function f is continuous, then one can obviously compute f by a CCT without any branching. Now let f be a function with $Lev(f) = n+1$ $(n \geq 1)$. By Lemma 3.2 the function $f|_{A_0 \setminus \overline{A_1}}$ is continuous. Furthermore the function $f|_{\overline{A_1}}$ has level n by Lemma 3.1 c) and by Lemma 3.2. Let by inductive assumption T_n be a CCT which computes $f|_{\overline{A_1}}$ using $n-1$ tests. We define the continuous function g_1 by $g_1(x) := -distance(x, \overline{A_1})$. Then $g_1(x) < 0 \iff x \in \overline{A_1}^c$. The following CCT T_{n+1} computes f using n tests, thus $Size(T_{n+1}) = n+1$:

$$T_{n+1} := (if\ g_1(x) < 0 \text{ then } (y := f|_{A_0 \setminus \overline{A_1}}(x)) \text{ else } T_n)$$

b) If $range(f)$ is discrete, then $\overline{A_i} \cap A_0 = A_i \cap A_0$ for all $i \geq 1$. The CCT of part a) proves the assertion. □

4 Examples

In the following we determine and estimate the levels of discontinuity of several simple functions and problems arising in computational geometry.

1) Define $f_2 : \mathbb{R}^2 \longrightarrow \{0,1,2\}$ by

$$f_2(x,y) := \begin{cases} 0 & \textit{if } x \leq 0 \\ 1 & \textit{if } x > 0 \textit{ and } y \leq 0 \\ 2 & \textit{if } x > 0 \textit{ and } y > 0\,. \end{cases}$$

Then $A_0 = \mathbb{R}^2$, $A_1 = \{(x,y) \in \mathbb{R}^2 \mid x = 0 \text{ or } x > 0 \text{ and } y = 0\}$, $A_2 = \{(0,0)\}$, $A_3 = \emptyset$. Thus $Lev(f_2) = 3$.

2) *Number of points:* The function

$$f_3^{(n)} : \mathbb{R}^n \longrightarrow \{1,\ldots,n\},$$
$$f_3^{(n)}(\overline{x}) := card\{x_1,\ldots,x_n\}$$

has level n. One easily checks that $A_m = \{\overline{x} \mid card\{x_1,\ldots,x_n\} \leq n-m\}$ for $0 \leq m \leq n-1$. We obtain $Lev(f_3^{(n)}) = n$.

3) *Sorting problems:* Let S_n be the set of all permutations of $\{1,\ldots,n\}$ and $M_n := \{\overline{z} \in \mathbb{C}^n \mid i \neq j \Rightarrow z_i \neq z_j\}$. For $z = x + i\cdot y, \hat{z} = \hat{x} + i \cdot \hat{y} \in M_n$ we define $z < \hat{z} :\Longleftrightarrow x < \hat{x}$ *or* $x = \hat{x}$ *and* $y < \hat{y}$. We assume $M_n \subseteq \mathbb{R}^{2n}$. Furthermore for a set $\mathcal{F}$ of functions we set $minLev(\mathcal{F}) := \min\{Lev(f) \mid f \in \mathcal{F}\}$. Two of the following sorting problems are not given by a single function but by a set of functions.

Sort real vectors:

$$f_4^{(n)} : \mathbb{R}^n \longrightarrow \mathbb{R}^n,$$

$(\forall \overline{x} \in \mathbb{R}^n)$ $f_4^{(n)}(\overline{x})$ is a permutation of $\overline{x}$ and $f_4^{(n)}(\overline{x})_1 \leq \ldots \leq f_4^{(n)}(\overline{x})_n$.

Give a sorting permutation for real vectors:

$$\mathcal{F}_5(n) := \{\, f : \mathbb{R}^n \longrightarrow S_n \mid (\forall \overline{x} \in \mathbb{R}^n)\; x_{f(\overline{x})(1)} \leq \ldots \leq x_{f(\overline{x})(n)}\}.$$

Sort complex vectors lexicographically:

$$f_6^{(n)} : M_n \longrightarrow M_n,$$

$(\forall \overline{z} \in M_n)$ $f_6^{(n)}(\overline{z})$ is a permutation of $\overline{z}$ and $f_6^{(n)}(\overline{z})_1 < \ldots < f_6^{(n)}(\overline{z})_n$.

Put complex vectors into welldefined order:

$$\mathcal{F}_7(n) := \{\, f : M_n \longrightarrow M_n \mid (\forall (z_1,\ldots,z_n) \in M_n)\; f(z_1,\ldots,z_n) \text{ is a permutation of } (z_1,\ldots,z_n) \textit{ and } (\forall \pi \in S_n)\; f(z_1,\ldots,z_n) = f(z_{\pi(1)},\ldots,z_{\pi(n)})\}.$$

Theorem 3 *For all* $n \geq 1$
$Lev(f_4^{(n)}) = 1$, *i.e.* $f_4^{(n)}$ *is continuous,*
$minLev(\mathcal{F}_5(n)) = n$,
$Lev(f_6^{(n)}) = n$,
$n \geq minLev(\mathcal{F}_7(n)) \geq n+1-\min\{D_p(n) \mid p \textit{ prime}\}$, *where* $D_p(n)$ *is the sum of the digits in the expansion of* n *in base* p.

The first three statements of the theorem can be proved by geometric considerations. The last one is a consequence of a theorem by Vassiliev ([18]). Namely, one can prove that $minLev(\mathcal{F}_7(n))$ is equal to the Schwarz genus ([18]) of the covering map $h_n : M_n \longrightarrow M_n/S_n$. The number $minLev(\mathcal{F}_7(n))$ is essentially the topological complexity in the sense of Smale of the problem how to determine the zeros of a complex polynomial ([15]). It is an open problem whether the estimates for $minLev(\mathcal{F}_7(n))$ can be improved.

4) *Number of segment intersections* ([14]): Let us denote a segment in the plane by a vector $s \in \mathbb{R}^4$ giving the coordinates of the endpoints. The set $X := \{(x_1,y_1,x_2,y_2) \in \mathbb{R}^4 \mid (x_1,y_1) \neq (x_2,y_2)\}$ consists of the proper line segments. For $j \in \{8,9\}$ we define

$$f_j^{(n)} :\subseteq \mathbb{R}^{4n} \longrightarrow \{0,\ldots,\frac{n\cdot(n-1)}{2}\} \quad \textit{by}$$

$f_j^{(n)}(s_1,\ldots,s_n) :=$ number of unordered intersecting pairs $\{s_k,s_l\}$ *with* $k \neq l$

where

$dom(f_8^{(n)}) := \{\, \overline{s} \in X^n \mid$ at most two segments s_i, s_j intersect in any point$\}$,
$dom(f_9^{(n)}) := \{\, \overline{s} \in X^n \mid$ no two segments s_i, s_j contain a common proper line segment$\}$.

Theorem 4 $Lev(f_8^{(n)}) = 2n-2$ *for* $n \geq 2$,
$4n+1 \geq Lev(f_9^{(n)}) \geq 4n-10$ *for* $n \geq 5$ *and*
$Lev(f_9^{(n)}) = 1 + \frac{n\cdot(n-1)}{2}$ *for* $1 \leq n \leq 4$.

The theorem illustrates that it is important to know whether one can exclude certain types of degeneracy or not.

It is clear that a degeneracy can occur only when an endpoint of a segment lies on another segment. Figure 4.1 shows how to obtain configurations $\overline{s}$ with $lev(f_9^{(n)},\overline{s}) \geq 4n-10$ for $n \geq 5$. The other estimate $4n+1 \geq Lev(f_9^{(n)})$ is a consequence of the fact that

$f_9^{(n)}$ has *analytic domains of continuity*, see below.

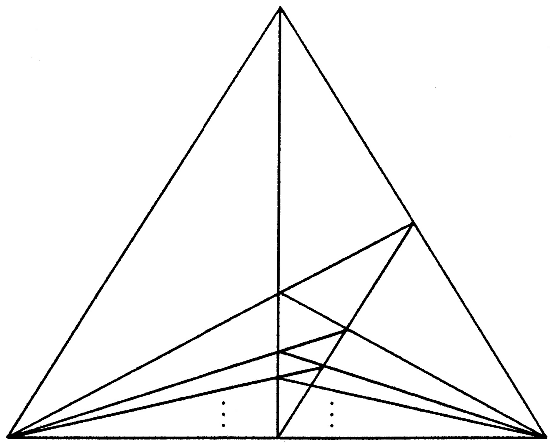

Figure 4.1

5) *Functions with analytic domains of continuity:* Usually a problem in computational geometry can be formalized by a partial function $f :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$. In most of the cases this function is constant or at least continuous on semialgebraic or semianalytic sets. Thus, it has analytic domains of continuity according to the following definition.

Definition 4.1 Fix a topological space T. A function $f :\subseteq \mathbb{C}^M \longrightarrow T$ (or $f :\subseteq \mathbb{R}^M \longrightarrow T$) has *analytic domains of continuity* iff for any point $z \in dom(f)$ there is a neighbourhood U of z and there are finitely many (real-)analytic functions $f_1, \ldots, f_k$ with $dom(f_j) = U$ such that f is continuous on the 2^k sets $\bigcap_{j=1}^k \tilde{N}_j$ where $\tilde{N}_j \in \{\{z \in U \mid f_j(z) = 0\}, \{z \in U \mid f_j(z) \neq 0\}\}$.

One easily checks that e.g. the function $f_9^{(n)}$ has analytic domains of continuity. We can prove that the level of functions as in Definition 4.1 is bounded by $M + 1$.

Theorem 5 *Let T be a topological space and let $f :\subseteq \mathbb{C}^M \longrightarrow T$ (or $f :\subseteq \mathbb{R}^M \longrightarrow T$) be a partial function with analytic domains of continuity. Then $Lev(f) \leq M + 1$.*

Usually in geometric problems M is a multiple of the size n of the problem.

5 Discussion of the computational problems caused by degeneracies

Usually algorithms for computational geometry problems are formulated in the real RAM model ([13]). In that model they are mathematically correct. But it seems to be very difficult to implement real RAM algorithms on physical computers (see e.g. [8]). This is an urgent problem as there are many real world applications which require fast and reliable geometric algorithms.

There are many heuristic approaches to solve the problem. For example one can try to discretize the problem and to work on a grid (see e.g. [5]). This makes it possible to use exact integer or rational arithmetic (see e.g. [16]). Another approach is to use symbolic data in addition to numeric data (see e.g. [9]). One might also change the input data slightly in order to obtain more stable configurations (see e.g. [11]). Or one could use varying error bounds (see e.g. [6]) or some other method. Some of the approaches are very promising. Nevertheless until now there is no mathematically precise realization of real RAM algorithms by physical computers. The main problem seem to be the tests $x < 0$, $g(x) = 0$ (g some computed function), etc., which lead to degenerate configurations.

Is there a theoretical reason why this problem is so difficult? In classical recursion theory Church's commonly accepted Thesis gives a mathematically precise notion of computable integer functions. A similar thesis for computable real functions has not yet been established in theoretical computer science. But there is a fundamental idea common to most of the theoretical approaches for defining "effectivity" in Analysis (see e.g. Bishop/Bridges [2], Troelstra/van Dalen [17], Grzegorczyk [4], Aberth [1], Weihrauch/Kreitz [20] or [19], Pour-El/Richards [12], Ko [10]) which we formulate here as a thesis.

Thesis 2 *Every intuitively or physically computable real function is continuous.*

The thesis, of course, requires some detailed discussion and justification. It has a very disillusioning consequence: It is impossible to compute discontinuous functions $f :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$, especially almost all geometric problems, (correctly) on physical devices. This is in accordance with the experience that it appears to be so difficult to implement geometric algorithms (correctly) on computers.

For a real RAM or CCT T the branchings are the only source of discontinuity of the computed function f. On the other hand by our results every function $f :\subseteq \mathbb{R}^M \longrightarrow \mathbb{R}^N$ of finite level can be computed by a CCT T such that the level of discontinuity is 1+ the number of branchings of T, and no CCT with less branchings computes f.

Since programs for geometric algorithms are needed in practice and since by Thesis 2 correct programs cannot exist in general, one can only try to find approximate implementations. Therefore mathematical definitions for "approximate" applyable to discontinuous functions and to implementations of real RAM's are needed and the possibility of implementing good approximations with low complexity has to be investigated. We believe that our concepts of level of discontinuity and size of a CCT can be useful for this purpose.

References

[1] Aberth, O.: *Computable analysis*, McGraw-Hill, New York (1980)

[2] Bishop, E., Bridges, D. S.: *Constructive analysis*, Springer-Verlag, Berlin, Heidelberg (1985)

[3] Blum, L., Shub, M. , Smale, S.: *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bull. Amer. Math. Soc. 21, 1 - 46 (1989)

[4] Grzegorczyk, A.: *On the definition of computable real continuous functions*, Fund. Math. 44, 61 - 71 (1957)

[5] Greene, D. H., Yao, F. F.: *Finite-resolution computational geometry*, Proc. of the 27th IEEE Symp. on Foundations of Comp..Sci, 143 - 152 (1986)

[6] Guibas, L., Salesin, D., Stolfi, J.: *Epsilon geometry: Building robust algorithms from imprecise computations*, Proc. of the 5th ACM Symp. on Computational Geometry, 208 - 217 (1989)

[7] Hertling, P.: *Topologische Komplexitätsgrade von Funktionen mit endlichem Bild*, Informatik-Berichte no. 152, Fernuniversität, Hagen (1993)

[8] Hoffmann, C. M.: *The problems of accuracy and robustness in geometric computation*, Computer, Vol. 22, no. 3, 31 - 42 (1989)

[9] Hoffmann, C. M., Hopcroft, J. E., Karasick, M. S.: *Towards implementing robust geometric computations*, Proc. of the 4th ACM Symp. on Computational Geometry, 106 - 117 (1988)

[10] Ko, K.-I: *Complexity Theory of Real Functions*, Birkhäuser, Boston (1991)

[11] Milenkovic, V.: *Verifiable implementations of geometric algorithms using finite precision arithmetic*, Artificial Intelligence 37, 377 - 401 (1988)

[12] Pour-El, M. B., Richards, J. I.: *Computability in Analysis and Physics*, Springer-Verlag, Berlin, Heidelberg (1988)

[13] Preparata, F. P., Shamos, M. I.: *Computational geometry*, Springer-Verlag, New York (1985)

[14] Shamos, M. I., Hoey, D.: *Geometric intersection problems*, Seventeenth Annual IEEE Symposium on Foundations of Computer Science, 208 - 215 (1976)

[15] Smale, S.: *On the topology of algorithms, I*, J. of Compl. 3, 81-89 (1987)

[16] Sugihara, K.: *An approach to error-free solid modeling*, IMA Summer Program on Robotics, Inst. Math. and Applications, Univ. of Minnesota, Minneapolis, Minn. (1987)

[17] Troelstra, A. S., van Dalen, D.: *Constructivism in mathematics*, Studies in Logic and the Foundations of Mathematics, Elsevier Science Publishers, North Holland, Amsterdam (1988)

[18] Vassiliev, V. A.: *Complements of discriminants of smooth maps: topology and applications*, Transl of math. monographs vol. 98, Am. Math. Soc., Providence, Rhode Island (1992)

[19] Weihrauch, K.: *Computability*, Springer-Verlag, Berlin, Heidelberg (1987)

[20] Weihrauch, K., Kreitz, Ch.: *A unified approach to constructive and recursive analysis*, in: Computation and proof theory (eds.: M. M. Richter et al.), Springer-Verlag, Berlin, Heidelberg (1984)

[21] Yap, C.-K.: *Symbolic treatment of geometric degeneracies*, J. Symbolic Computation 10, 349 - 370 (1990)