

# Computable Partial Solids and Voxels Sets

André Lieutier

Scientific team, Matra Datavision & XAOLAB, LIM, Marseilles, France

**Abstract.** The model of computable partial solids has been recently introduced in order to address computational geometry and solid modeling issues within the Turing model of computation. This approach provides a model that reflects well the observable properties of real solids and the computation on realistic computers [5]. Since a central notion of discrete geometry, voxel sets, can be used to define computable partial solids, this approach throws a bridge between discrete geometry and solid modeling in  $\mathbf{R}^n$ . This paper presents this model and the recursive analysis and domain theory prerequisites.

## 1 Introduction

Discrete geometry is used to model “continuous geometry”. Discrete geometry involves finite data only and can then be handled by realistic computers, or, more theoretically, by Turing equivalent (for the sake of computability) computers. This situation is quite common in practical computations. Elements of countable sets are used to model or approximate elements of uncountable sets whose representation would involve infinite data.

The availability of various data type modeling real numbers (floating point as well as more elaborate real numbers arithmetics) has led to design algorithms assuming a machine model, the “Real RAM”, able to perform exact real numbers computations [12,2]. This model of computation is realistic as long as the computed operators are continuous, since discontinuous operators as the real numbers comparison predicate are not Turing computable [11]. Actually, practical implementations of geometric algorithms which correctness have been proved under the assumption of the real RAM machine model often suffer of so called “robustness problems”.

To address computational geometry and geometric modeling computations with the Turing machine model, one disposes of existing mathematical models of computation over uncountable sets, as recursive analysis and domain theory. In this context, a model for representing solids, the *solid domain*, has been built using rational polyhedra or voxels sets [9,10,5]. This model connects geometry of solids in  $\mathbf{R}^n$  with discrete geometry.

In section 2, the properties of computation over uncountable sets are introduced informally in order to motivate the point of view of recursive analysis. In section 3, some elements of recursive analysis and domain theory used in section 4 are outlined. Section 4 presents the solid domain and some properties of computations over solids due to a common work with Pr. **Abbas Edalat** [6].

## 2 Computation over uncountable sets

Actual computers deal with finite data only. Each element of a countable set, as an integer or a rational number for example, can be represented by a finite word over a finite alphabet. Computing such an element means computing its finite representation. Because finite length words can represent elements of countable sets only, computation over elements of uncountable sets needs a more elaborate definition.

Let us consider as first example the number  $\pi$ . Even if it belongs to the uncountable set of real numbers, it is reasonable to say that there exists programs able to compute it. In this case, what one asks from such a program is to be able to compute an arbitrary accurate approximation of  $\pi$ . For example, a program that, given any integer  $i$  as input, computes a dyadic number  $d_i$  such that  $|\pi - d_i| < 2^{-i}$  is said to compute  $\pi$  (a *dyadic number* is a rational number whose denominator is a power of 2).

Let us take as second example the computation of the solution of a PDE (for Partial Differential Equation) by a Ritz method. The weak form of a PDE is a minimization problem in an infinite dimensional real (or complex) vector space  $F$  of functions. Numerical methods use a sequence  $s = \{s_i, i \in \mathbf{N}\}$  such that finite linear combinations of elements of  $s$  are dense in  $F$ . An algorithm is said to compute the solution if it is able to compute, given any integer  $i$  as input, an up to  $2^{-i}$  approximation of the solution for a given metric<sup>1</sup>. The approximations are finite linear combinations of elements of  $s$ . In actual computations, the coefficients of these finite linear combinations of elements of  $s$  are often floating point numbers. From a theoretical point of view, it is enough to assume that these coefficients belong to a dense subset of  $\mathbf{R}$  as for example dyadic numbers.

We can understand with this two simple examples two major ingredients of computation of an element of an uncountable set  $E$  :

- because computing such an element means computing it up to an arbitrary accuracy, one see that computability over uncountable sets depends on a metric or at least a topology on  $E$ .
- because approximations have to be represented by finite words, they have to belong to a countable set  $C$  and, to avoid a priori restriction in the set of approximable elements, it is natural to ask from  $C$  to be dense in  $E$ .

The computations of approximations of subsets of  $\mathbf{R}^n$  by subsets of  $\mathbf{N}^n$ , or voxel sets, is still another example, addressed in section 4.

With this point of view of computation, an operator  $O$  is computable if there exists a program that, given any input  $I$ , is able to compute an arbitrary accurate approximation of  $O(I)$  using an arbitrary accurate approximation of  $I$ . This informal formulation of computability shows the important proposition **“computable implies continuous”**. This point of view is formalized in the next section.

<sup>1</sup> The “right” metric depends on the nature of the PDE. For elliptic PDE, the set  $F$  may be a Sobolev space with its associated metric. Within finite elements methods, for example, the sequence  $s$  is usually made of piece-wise polynomial functions basis.

### 3 Recursive analysis and domain theory

In this section, we give a few definitions used in subsequent sections.

#### 3.1 Recursion theory

*Computable functions* from  $\mathbf{N}$  to  $\mathbf{N}$  are the partial functions that can be computed by a Turing machine<sup>2</sup>. We use the equivalent notion of *partial recursive functions* from  $\mathbf{N}$  to  $\mathbf{N}$  as a map from  $\mathbf{N}$  to  $\mathbf{N}$  that can be computed by a general purpose computer or a (finite length) program written in a general purpose language. These functions are *partial* since, for some integer values given as inputs, the program may run for ever. We say that a partial recursive function is *recursive* if it is total, that is if it ends within a finite number of steps of computation for each integer given as input.

A subset of  $\mathbf{N}$  is said *r.e.* (for *recursively enumerable*) if it is empty or if it is the range (that is the image set) of a recursive function. A subset  $A$  of  $\mathbf{N}$  is r.e. if and only if there exists a partial recursive function  $f_A$  such that  $f_A(i)$  is defined and equal to 1 if and only if  $i \in A$ .

A subset of  $\mathbf{N}$  is said *recursive* if itself and its complement are r.e.. A subset  $A$  of  $\mathbf{N}$  is recursive if and only if there exists a recursive function  $f_A$  such that  $f_A(i)$  equal 1 for  $i \in A$  and 0 for  $i \notin A$ . There are r.e. sets that are not recursive, but their construction is not trivial.

Because there exists computable, one-to-one maps between  $\mathbf{N}$  and  $\mathbf{N}^k$  for each  $k$ , the definitions above (partial recursive and recursive maps, r.e. and recursive subsets) extends naturally to maps from  $\mathbf{N}^k$  to  $\mathbf{N}^{k'}$  and to subsets of  $\mathbf{N}^k$ . These notions apply similarly to countable sets.

#### 3.2 Recursive analysis

Recursive analysis [11] could be defined as the extension of recursion theory to computation over uncountable sets. Since  $\mathbf{Q}$ , the set of rational numbers, is countable, it is in one-to-one correspondence with  $\mathbf{N}$  and we can write  $\mathbf{Q} = \{q_n | n \in \mathbf{N}\}$  where  $q_n = q_m$  if and only if  $n = m$ . A *real number  $x$  is said computable* if and only if there exists a recursive function  $f$  such that  $\forall n \in \mathbf{N}, |x - q_{f(n)}| \leq 2^{-n}$ <sup>(3)</sup>. Since the set of recursive functions, as the set of finite programs, is countable, there are non computable real numbers.

In a metric space, a sequence  $(x_n)_{n \in \mathbf{N}}$  is said to converge *effectively* toward  $x$  if and only if there exists a recursive function  $f$  such that

$$\forall n, m \in \mathbf{N}, m \geq f(n) \Rightarrow d(x, x_m) \leq 2^{-n}$$

Note that, in the definition of a computable real number, one do not ask only for the existence of a computable sequence of rational numbers converging to  $x$ :

<sup>2</sup> See the Gödel-Kleene theorem [1].

<sup>3</sup> This definition has been given by Turing in 1937.

the convergence have to be effective. A real number is said *lower (respectively upper) semi-computable* if and only if it is the limit of an increasing (respectively decreasing) computable sequence of rational numbers. A real number is computable if and only if it is lower semi-computable and upper semi-computable.

A function  $f$  from  $[0, 1]$  to  $\mathbf{R}$  is said *computable* if it is the effective limit, for the *sup* norm, of a computable sequence of polynomials with rational coefficients. It can be shown that an equivalent definition is that there exists a program able, for any real  $x$  in  $[0, 1]$  and any integer  $i$  given as input, to compute :

- an integer  $j$ ,
- an up to  $2^{-i}$  approximation of  $f(x)$  using an up to  $2^{-j}$  approximation of  $x$ .

It follows that such a function is continuous.

A general theory of computability over uncountable sets has been developed [16], based on a modified Turing machine, the *TTE* (for Type Two theory of Effectivity), whose inputs and output are infinite sequences of symbols. The solid domain and other notions introduced here could be defined in this framework (see [3]). However, we use hereafter a domain-theoretic framework because, even if it is, in some sense, equivalent, domain theoretic approach highlights the links between computability and general topology. It allows to build up most of the mathematical structures in the framework of partial orders and general topology, introducing lately the “Turing computability ingredient” through the notions of r.e. subsets of  $\mathbf{N}$  and recursive functions.

### 3.3 Domain theory

Domain theory was originally introduced independently by Scott [15] as a mathematical theory of semantics of programming languages and by Ershov [4] for studying partial computable functionals of finite type.

A domain is a mathematical structure for data types or operators representing incomplete or uncertain information. It is a partially ordered set  $(D, \sqsubseteq)$  where the partial order corresponds to some notion of information. “ $A \sqsubseteq B$ ” means “the information represented by A is contained in the information represented by B”.

A simple example is the domain  $\{\text{tt}, \text{ff}, \perp\}$  of the Boolean values *tt* and *ff* together with a least element  $\perp$  below both. One thinks of  $\perp$  here as the undefined Boolean value that represents “no information at all”.

Another example is the *interval domain*  $\mathbf{I}[0, 1]$ . It is the set of all non empty real intervals  $[a, b]$ , with  $0 \leq a \leq b \leq 1$ . The information represented by  $[a, b]$  can be seen as the information concerning a real value  $x$ :  $x \in [a, b]$ . The information order  $\sqsubseteq$  is then the reverse inclusion order :  $[a, b] \sqsubseteq [a', b'] \iff [x \in [a', b'] \Rightarrow x \in [a, b]] \iff [a', b'] \subset [a, b] \iff a \leq a'$  and  $b' \leq b$ . There is a bottom element  $\perp$  in  $\mathbf{I}[0, 1]$ , that is an element below any other:  $\perp = [0, 1]$ .

**Formal definitions.** Let us consider a partial order  $(P, \sqsubseteq)$ . A subset  $\Delta$  of  $P$  is said *directed* if and only if :

$$\forall x, y \in \Delta, \exists z \in \Delta, x \sqsubseteq z \text{ and } y \sqsubseteq z$$

With the interpretation of  $\sqsubseteq$  as an information order, a directed subset is a set of consistent values: given two values  $x$  and  $y$  of the subset, the information represented by  $x$  and  $y$  is consistent, since it is refined by a value  $z$  in the set.

A *directed complete partial order*, or *dcpo* is a partial order such that each directed set has a *lower upper bound*, or *lub*. With the interpretation of  $\sqsubseteq$  as an information order, in a dcpo the information contained in any directed set can be represented by an element of the dcpo, the lub of the subset. dcpo's are also called *domains*.

There is a  $T0^4$  topology associated to dcpo's: *the Scott topology*. A subset  $O$  of a dcpo  $(D, \sqsubseteq)$  is *Scott open* if and only if :

- It is upward closed:  $x \in O$  and  $x \sqsubseteq y \Rightarrow y \in O$ ,
- and if it contains the lub  $\delta$  of a directed subset  $\Delta$ , it contains an element of  $\Delta$ :  $\delta \in O \Rightarrow \exists x \in \Delta, x \in O$

This topology defines continuous functions among dcpo's or among dcpo's and topological spaces. One can show that a map  $f$  between dcpo's is continuous (that is Scott-continuous) if and only if  $f$  is monotonic and  $f$  preserves lub's, that is, the lub of the image by  $f$  of a directed set  $\Delta$  is the image by  $f$  of the lub of  $\Delta$ .

For example, one can check that the two examples above  $(\{\mathbf{tt}, \mathbf{ff}, \perp\})$  and  $\mathbf{I}[0, 1]$  are dcpo's. Open sets of  $\{\mathbf{tt}, \mathbf{ff}, \perp\}$  are  $\emptyset$ ,  $\{\mathbf{tt}\}$ ,  $\{\mathbf{ff}\}$ , and  $\{\mathbf{tt}, \mathbf{ff}, \perp\}$ .

The set of all the  $[a, b]$  intervals included in a given open<sup>5</sup> subset  $O$  of  $[0, 1]$  is a Scott open subset of  $\mathbf{I}[0, 1]$ . The set of all such sets is a basis of the (Scott-)open sets of  $\mathbf{I}[0, 1]$ .

Given two elements  $x, y$  in a dcpo  $D$ , we say that  $x$  is *way below*  $y$ , or  $x$  *approximates*  $y$ , denoted by  $x \ll y$  if, for any directed subset  $\Delta$  with a lub  $\delta$ :  $y \sqsubseteq \delta \Rightarrow \exists z \in \Delta, x \sqsubseteq z$ . One can check that, in  $\mathbf{I}[0, 1]$ ,  $[a, b] \ll [a', b'] \iff [a', b'] \subset ]a, b[ \iff a < a'$  and  $b' < b$ .

An element  $x$  of a dcpo  $D$  is said *maximal* if there is no other element above  $x$ :  $\forall y \in D, x \sqsubseteq y \Rightarrow x = y$ . For examples, the maximal elements of  $\{\mathbf{tt}, \mathbf{ff}, \perp\}$  are  $\mathbf{tt}$  and  $\mathbf{ff}$ . The maximal elements of  $\mathbf{I}[0, 1]$  are the  $[x, x]$  intervals. The set of maximal elements of  $\mathbf{I}[0, 1]$  can then be identified to  $[0, 1]$ .

A continuous domain [7] is a dcpo  $(D, \sqsubseteq)$  such that, for every element  $x \in D$ :

- the set  $X = \{y \in D \mid y \ll x\}$  is directed,
- and  $x$  is the lub of  $X$ .

It can be shown that  $\mathbf{I}[0, 1]$  is a continuous domain. A subset  $B$  of a continuous domain  $(D, \sqsubseteq)$  is a *basis* of  $D$  if, for each element  $x$  of  $D$ :

- the set  $B_x = \{b \in B \mid b \ll x\}$  is directed,
- and  $x$  is the lub of  $B_x$ .

<sup>4</sup> A  $T0$  topology is a topology that satisfies the weakest separation axiom.

<sup>5</sup> Open for the topology induced by the standard topology of  $\mathbf{R}$ .

An  $\omega$ -continuous domain is a continuous domain with a countable basis. An element of an  $\omega$ -continuous domain can be expressed as the lub of a sequence of basis elements approximating it. One can see the relation with the informal description of computation over uncountable sets given in section 2.  $\mathbf{I}[0, 1]$  is an  $\omega$ -continuous domain. Examples of countable basis are the sets of intervals  $[p, q]$  where  $p$  and  $q$  are rational or dyadic numbers.

### 3.4 Computability over effectively given continuous domains

Recently,  $\omega$ -continuous domains has been used to define computability structures [7,8].

We say that an  $\omega$ -continuous domain  $(D, \sqsubseteq)$  with a basis  $B$  is *effectively given* with respect to a given enumeration of  $B = \{b_0, b_1, b_2, \dots\}$ , with  $b_0 = \perp$ , if  $\{(i, j) \mid b_i \ll b_j\}$  is an r.e. subset of  $\mathbf{N}^2$ .

We say that an element  $x$  of  $D$  is *computable* if the set  $\{i \mid b_i \ll x\}$  is r.e.. This is equivalent to say that there exists a recursive function  $f$  such that the sequence of  $(b_{f(i)})_{i \in \mathbf{N}}$  is increasing and  $x$  is the lub of the sequence. An element  $[a, b]$  of  $\mathbf{I}[0, 1]$  is computable if and only if  $a$  is lower semi-computable and  $b$  is upper semi-computable. A maximal element  $[x, x]$  is computable if and only if  $x$  is a computable real number.

A map  $f$  between effectively given  $\omega$ -continuous domains  $D$  and  $E$  with the respective bases  $\{d_0, d_1, d_2, \dots\}$  and  $\{e_0, e_1, e_2, \dots\}$  is computable if the set  $\{(i, j) \mid e_i \ll f(d_j)\}$  is an r.e. subset of  $\mathbf{N}^2$ .

### 3.5 $\mathbf{I}[0, 1]^n$

The *interval domain*  $\mathbf{I}[0, 1]^n$  of  $[0, 1]^n$  is the set of all non-empty  $n$ -dimensional sub-rectangles in  $[0, 1]^n$  ordered by reverse inclusion. It is an  $\omega$ -continuous domain. A basic Scott open set is given, for every open subset  $O$  of  $\mathbf{R}^n$ , by the collection of all rectangles contained in  $O$ . The set of maximal elements is the set of  $\{x\}$  for  $x \in [0, 1]^n$ . The set of sub-rectangles with dyadic coordinates provides a countable basis of  $\mathbf{I}[0, 1]^n$ . A basis element of  $\mathbf{I}[0, 1]^n$  may represent a voxel (or a rectangular set of voxels).

## 4 The solid domain

### 4.1 Continuity

For any subset  $A$  of a topological space,  $\overline{A}$ ,  $A^\circ$ ,  $\partial A$  and  $A^c$  denote respectively the closure, the interior, the boundary and the complement of  $A$ . The *regularization* of a subset  $A$  is defined, by Requicha [13,14], as the subset  $\overline{A^\circ}$ . We say that a set is *regular* if it is equal to its regularization.

The *solid domain*  $\mathbf{S}[0, 1]^n$  of the unit cube  $[0, 1]^n \subset \mathbf{R}^n$  is the set of ordered pairs  $(A, B)$  of compact subsets of  $[0, 1]^n$  with  $A \cup B = [0, 1]^n$  endowed with the information order:  $(A_1, B_1) \sqsubseteq (A_2, B_2) \iff A_2 \subset A_1 \text{ and } B_2 \subset B_1$ . The elements of  $\mathbf{S}[0, 1]^n$  are called *partial solids*.

**Proposition 1.**  $(\mathbf{S}[0, 1]^n, \sqsubseteq)$  is a continuous domain.

The meaning of the way below or approximation relation between partial solids is then given by the following proposition:

**Proposition 2.** For any  $(A, B)$  and  $(A', B')$  in  $\mathbf{S}[0, 1]^n$

$$(A, B) \ll (A', B') \iff A' \subset A^\circ \text{ and } B' \subset B^\circ$$

The next proposition shows that the solid domain could be equivalently defined as the quotient of the power set of  $[0, 1]^n$  under the equivalence relation  $X \equiv Y \iff (X^\circ = Y^\circ \ \& \ \overline{X} = \overline{Y})$  equipped with the corresponding ordering.

**Proposition 3.** For any  $(A, B) \in \mathbf{S}[0, 1]^n$ , there exists a subset  $Y$  of  $[0, 1]^n$  such that:  $A = \overline{Y}$  and  $B = \overline{Y^c}$ .

One can take for example:  $Y = (A \setminus B) \cup \partial A \cup (\mathbf{Q}^n \cap (A \cap B)^\circ)$ .

Given any subset  $X$  of  $[0, 1]^n$ , the classical membership predicate  $\in: [0, 1]^n \rightarrow \{\text{tt}, \text{ff}\}$  is continuous except on  $\partial X$ . It follows that the best continuous approximation of this predicate is  $\in': [0, 1]^n \rightarrow \{\text{tt}, \text{ff}, \perp\}$  where the value  $\perp$  is taken on  $\partial X$  (recall that any open set containing  $\perp$  contains the whole  $\{\text{tt}, \text{ff}, \perp\}$  set). Conversely, any continuous function from  $[0, 1]^n$  to  $\{\text{tt}, \text{ff}, \perp\}$  is the membership predicate of a partial solid. The solid domain could then be equivalently defined as  $\mathbf{C}([0, 1]^n, \{\text{tt}, \text{ff}, \perp\})$ , the set of continuous maps from  $[0, 1]^n$  to  $\{\text{tt}, \text{ff}, \perp\}$  with the ordering induced by the order of  $\{\text{tt}, \text{ff}, \perp\}$  [5,6]. The following proposition connects the Solid domain to the notion of Requicha’s regular sets.

**Proposition 4.** The maximal elements of  $\mathbf{S}[0, 1]^n$  are precisely those that represent regular sets. In other words, maximal elements are of the form  $(A, B)$  such that  $A$  and  $B$  are regular with  $B = \overline{A^c}$  and  $A = \overline{B^c}$ .

**Hausdorff and Lebesgues metrics.** Two metrics over partial solids are used in subsequent sections. The Hausdorff metric  $d_K$  between compact sets is a natural metric for the purpose of approximation of compact sets: recall that the set  $\mathbf{K}([0, 1]^n)$  of compact subsets of  $[0, 1]^n$  equipped with  $d_K$  is a complete metric space (that is, any Cauchy sequence is converging). The metric induced on  $\mathbf{S}[0, 1]^n$  as a subset of  $[\mathbf{K}([0, 1]^n)]^2$ :

$$d_H((A, B), (A', B')) = \max(d_K(A, A'), d_K(B, B'))$$

is also called (by an abuse of language) the Hausdorff metric over  $\mathbf{S}[0, 1]^n$ . Because  $\mathbf{S}[0, 1]^n$  is a closed subset of  $[\mathbf{K}([0, 1]^n)]^2$  for the topology defined by the Hausdorff metric,  $\mathbf{S}[0, 1]^n$  is a complete metric space for  $d_H$ .

Now, given a compact subset  $X$  of  $[0, 1]^n$ , the characteristic function of  $X$ , that is the function whose value is one on  $X$  and zero elsewhere, is denoted  $\chi_X$ . The  $L^1$  norm of  $\chi_X$  is the Lebesgues measure of  $X$ . The following semi-metric  $d_L$ , defined by

$$d_L(X, Y) = \| \chi_X - \chi_Y \|_1$$

is the Lebesgues measure of the set theoretic symmetric difference between  $X$  and  $Y$ . It is not a metric as the Lebesgues semi-metric between two compact sets that differ by a zero measure set is zero. This semi-metric induces a semi-metric on  $\mathbf{S}[0, 1]^n$  that is called Lebesgues semi-metric here. It defines of course a metric on the associated quotient set.

**Boolean operators and generalized membership predicate.** The truth tables of logical Boolean operators on  $\{\text{tt}, \text{ff}, \perp\}$  give us the definition of Boolean operators on the solid domain as we want to keep the natural definition of union:  $x \in X \cup Y \iff x \in X \text{ or } x \in Y$ . Applying the same rule for intersection and complement give the definition of Boolean operators on partial solids:

$$\begin{aligned} (A_1, B_1) \cup (A_2, B_2) &= (A_1 \cup A_2, B_1 \cap B_2) \\ (A_1, B_1) \cap (A_2, B_2) &= (A_1 \cap A_2, B_1 \cup B_2). \\ (A, B)^c &= (B, A). \end{aligned}$$

The union operator over  $\mathbf{K}([0, 1]^n)$  is continuous for the Hausdorff metric while the intersection operator is not. The union and intersection operators are continuous (even Lipchitzian) for the Lebesgues metric over (the quotient of)  $\mathbf{K}([0, 1]^n)$ .

We have then the following properties:

**Proposition 5.** *The  $\cup$  and  $\cap$  operators over  $\mathbf{S}[0, 1]^n$  are not continuous for the Hausdorff metric.*

**Proposition 6.** *The  $\cup$  and  $\cap$  operators over the quotient of  $\mathbf{S}[0, 1]^n$  are continuous for the Lebesgues metric.*

We have defined the continuous membership predicate for points of  $[0, 1]^n$ . In order to be able to compute this predicate, we extend it to the interval domain  $\mathbf{I}[0, 1]^n$  by defining  $\in: \mathbf{I}[0, 1]^n \times \mathbf{S}[0, 1]^n \rightarrow \{\text{tt}, \text{ff}, \perp\}$  with:

$$\in (X, (A, B)) = \begin{cases} \text{tt} & \text{if } X \subset B^c \\ \text{ff} & \text{if } X \subset A^c \\ \perp & \text{otherwise} \end{cases}$$

**Proposition 7.** *The following maps are continuous for the Scott topology, in other words they are monotonic and preserve lower upper bounds:*

$$\begin{aligned} \in: \mathbf{I}[0, 1]^n \times \mathbf{S}[0, 1]^n &\rightarrow \{\text{tt}, \text{ff}, \perp\} \\ \cup: \mathbf{S}[0, 1]^n \times \mathbf{S}[0, 1]^n &\rightarrow \mathbf{S}[0, 1]^n \\ \cap: \mathbf{S}[0, 1]^n \times \mathbf{S}[0, 1]^n &\rightarrow \mathbf{S}[0, 1]^n \end{aligned}$$



## 4.2 Computability

In this section, we introduce three different notions of computable partial solids related to the topologies defined in the previous sections. These three notions of computable partial solids lead to corresponding notions of computable operators over partial solids.

To define computability, we need a countable basis of  $\mathbf{S}[0, 1]^n$ . A *dyadic voxels set* is a finite union of cubes, each the product of  $n$  intervals whose end points are dyadic numbers. A *partial dyadic voxels set* ( $pdvs$ ) is an element  $(A, B) \in \mathbf{S}[0, 1]^n$  such that  $A$  and  $B$  are dyadic voxels sets. **PDVS** stands for the set of  $pdvs$ 's. The set **PDVS** is effectively enumerable. This means that each  $pdvs$  can be represented by a finite word and there exists a program able to compute a one-to-one correspondance between  $\mathbf{N}$  and **PDVS** so that one can denote by  $pdvs_i$  the  $i^{th}$   $pdvs$  and  $\mathbf{PDVS} = \{pdvs_i, i \in \mathbf{N}\}$ .

Every compact subset of  $[0, 1]^n$  is the intersection of all the dyadic voxels set containing it. It follows that every partial solid is the least upper bound of all the  $pdvs$  below it. In other words, we have the following proposition:

**Proposition 8.** *PDVS is a countable basis of the solid domain  $\mathbf{S}[0, 1]^n$ .*

Moreover, there exists a program able to decide, given two integers  $i$  and  $j$  as input, if  $pdvs_i \ll pdvs_j$  or not. This means that the set  $\{(i, j) \mid pdvs_i \ll pdvs_j\}$  is a recursive subset of  $\mathbf{N}^2$ . It is then a fortiori recursively enumerable. This last statement means the following:

**Proposition 9.** *The solid domain is effectively given with respect to the enumeration  $\{pdvs_i, i \in \mathbf{N}\}$  of **PDVS**.*

It is now possible to introduce the three notions of computability. A partial solid  $(A, B)$  is said *computable* if it is computable according to the general definition given in section 3 for maps between effectively given continuous domains, that is, if the set  $\{k \mid pdvs_k \ll (A, B)\}$  is recursively enumerable.

It is said *recursive* if the set  $\{k \mid pdvs_k \ll (A, B)\}$  is recursive.

It is said *Lebesgues computable* if it is computable and the Lebesgues measures of  $A$  and  $B$  are computable real numbers.

In [3], several related notions of computability for compact sets are given in the TTE framework.  $(A, B)$  is a computable partial solid if  $A$  and  $B$  are c.r.e. compact sets according to the terminology of [3], and  $(A, B)$  is recursive if  $A$  and  $B$  are recursive compact sets according to the terminology of [3]. The following propositions state other equivalent definitions for computable, recursive and Lebesgues computable partial solids. Herefter,  $pdvs_i$ , the  $i^{th}$   $pdvs$ , is denoted  $(A_i, B_i)$ .

**Proposition 10.** *The following statements are equivalents:*

- *The partial solid  $(A, B)$  is computable.*
- *There exists a recursive function  $f$  such that  $A = \bigcap_k A_{f(k)}$ ,  $B = \bigcap_k B_{f(k)}$ .*
- *There exists a recursive function  $g$  such that  $A = \bigcap_k A_{g(k)}$ ,  $B = \bigcap_k B_{g(k)}$  and the sequence of  $(A_{g(k)}, B_{g(k)})$  is increasing for the information order  $\sqsubseteq$ .*

In the situation of the third statement of the last proposition, the sequence  $(A_{g(k)}, B_{g(k)})$  converges toward  $(A, B)$  for the Hausdorff metric. The proposition below (that can be shown using [3]) states that this convergence is effective if and only if  $(A, B)$  is recursive.

**Proposition 11.** *The following statements are equivalents:*

- The partial solid  $(A, B)$  is recursive.
- There exists a recursive function  $f$  such that  $A = \bigcap_k A_{f(k)}$ ,  $B = \bigcap_k B_{f(k)}$  and  $d_H((A, B), (A_{f(k)}, B_{f(k)})) \leq 2^{-k}$ .

We have a similar proposition for Lebesgues computability. Here, for a compact subset  $X$  of  $[0, 1]^n$ ,  $\mu(X)$  denotes the Lebesgues measure of  $X$ .

**Proposition 12.** *The following statements are equivalents:*

- The partial solid  $(A, B)$  is Lebesgues computable.
- There exists a recursive function  $f$  such that  $A = \bigcap_k A_{f(k)}$ ,  $B = \bigcap_k B_{f(k)}$  and the sequence  $(A_{f(k)}, B_{f(k)})$  converges effectively toward  $(A, B)$  for the Lebesgues (semi)-metric.
- There exists a recursive function  $g$  such that  $A = \bigcap_k A_{g(k)}$ ,  $B = \bigcap_k B_{g(k)}$  and  $\mu(A_{g(k)} \cap B_{g(k)}) - \mu(A \cap B) \leq 2^{-k}$ .

**Computability of the boolean operators and the membership predicate.** From the effectively enumerated basis  $\{pdvs_i, i \in \mathbf{N}\}$  of  $\mathbf{S}[0, 1]^n$  one can define an effectively enumerated basis of the product domain  $\mathbf{S}[0, 1]^n \times \mathbf{S}[0, 1]^n$  that is then an effectively given domain.

**Proposition 13.** *Using the effectively enumerated bases given above, the  $\cup$  and  $\cap$  operators are computable maps from  $\mathbf{S}[0, 1]^n \times \mathbf{S}[0, 1]^n$  to  $\mathbf{S}[0, 1]^n$ .*

In other words, there exists a program that, given any pair of computable partial solids given by the computable sequences of  $pdvs$  approximating them, computes a sequence of  $pdvs$ 's approximating their union (respectively their intersection). Because each element of the output sequence is computed in a finite time, each of these computations use finitely many elements of the input sequences.

**Proposition 14.** *Using the effectively enumerated bases given above, the  $\in$  predicate is a computable map from  $\mathbf{I}[0, 1]^n \times \mathbf{S}[0, 1]^n$  to  $\mathbf{S}[0, 1]^n$ .*

From the fact that boolean operators are not continuous for the Hausdorff metric, it follows that they can not satisfy the corresponding notion of computability. Moreover, we have the following:

**Proposition 15.** *They are pairs of recursive partial solids whose union (respectively intersection) is not recursive.*

This means that the set of recursive solids is not closed under boolean operators. On another hand, the information given by the computable sequence defining a computable solid is rather weak as, given any finite part of the sequence, one do not know how far is the actual solid.

However, for Lebesgues computable partial solid there is a computable sequence that approximate the partial solid up to any given accuracy for the Lebesgues semi-metric.

Fortunately, we have:

**Proposition 16.** *Boolean operators are Lebesgues computable.*

In other words, there exists a program that, given any pair of Lebesgues computable partial solids given by the computable sequences of pdvs's approximating them and effectively converging for the Lebesgues metric, computes a sequence of pdvs's approximating their union (respectively their intersection) effectively converging for the Lebesgues metric. The computation of each element of the output sequence use finitely many elements of the input sequences. The second and third statements of proposition 12 give examples of a such sequences of pdvs's approximating a partial solid and effectively converging for the Lebesgues metric.

## 5 Applications and further developpements

The primary goal of this approach is the design of robust geometric algorithms. Robustness issues show up when the values of the logical predicates evaluated from numerical computations are inconsistent, resulting in an invalid output or the catastrophic failure of the algorithm.

Arithmetics for real numbers data type hyde the uncountable nature of computations as far as one deals with continuous maps and operators. Evaluation of geometric predicates, as the point membership to a solid represented by its explicit boundary or by an implicit equation, is a typical example of discontinuous map. In order to make the mathematical model consistent with a realistic computation, domain theoretic approach considers continuous predicates valued in the boolean domain  $\{\text{tt}, \text{ff}, \perp\}$ . See [5] for an application to a completely specified and robust disk-disk intersection algorithm.

The partial solid model may also help to design realistic specifications of algorithms computing boolean operators over B.Rep<sup>6</sup> models. For example, it seems to be the right framework to define computation over subsets of  $[0, 1]^n$  defined by a implicit equation as:

$$E = \{x \in [0, 1]^n \mid f(x) \leq 0\}$$

For a computable continuous function  $f$  from  $[0, 1]^n$  to  $\mathbf{R}$ .

---

<sup>6</sup> B.Rep stand for Boundary Representation.

## 6 Acknowledgements

Original contents in this paper are the result of our common work with **Abbas Edalat** [6].

## References

1. Amadio, R.M.,Curien, P.L., Domains and Lambda-Calculi, Cambridge Tracts in Theoretical Computer Science, 1998.
2. Blum, L., Cucker, F., Shub, M., Smale, S. Complexity and Real Computation, Springer-Verlag (1998)
3. Brattka, V., Weihrauch, K. Computability on subsets of Euclidean Space I: Closed and Compact Subsets To appear in TCS, 1998 [www.informatik.fernuni-hagen.de/cca/](http://www.informatik.fernuni-hagen.de/cca/)
4. Y. L. Ershov, Y.L., Computable functionals of finite types, journal of Algebra and Logic, volume 11, number 4, pages 367-437, 1972.
5. Edalat, A., Lieutier, A. Foundation for a Computable Solid Modeling, submitted to ACM Solid Modeling 1999, <http://theory.doc.ic.ac.uk:80/ae/>.
6. Abbas Edalat, private mail. Departement of Computing, Imperial College, London. <http://theory.doc.ic.ac.uk:80/ae/>.
7. Edalat, A. Domains for computation in mathematics, physics and exact real arithmetic, Bulletin of Symbolic Logic, volume 3, number 4, page 401-452, 1997.
8. Edalat, A., Potts, P.J., A new representation for exact real numbers, Electronic Notes in Theoretical Computer Science, volume 6, Proceedings of Mathematical Foundations of Programming Semantics 13, 1997, Available from URL:- <http://www.elsevier.nl/locate/entcs/volume6.html>, Elsevier Science B. V.
9. Lieutier, A. Représentation B.Rep et calculabilité, Journées modeleurs géométriques (1997) LMC, IMAG, Université Joseph Fourier, BP 53X 38041 Grenoble Cedex, France.
10. Lieutier, A. Toward a data type for Solid Modeling based on Domain Theory, Workshop on Computability and Complexity in Analysis, (Brno 98) Informatik Berichte 235, FernUniversität Hagen, [www.informatik.fernuni-hagen.de/cca/](http://www.informatik.fernuni-hagen.de/cca/).
11. Pour-El, M.B., Richards, J.I., Computability in Analysis and Physics, Springer-Verlag (1989).
12. Preparata, F., Shamos, M, Computational Geometry: an introduction", Springer Verlag (1985).
13. Requiha, A., Tilove, R., Mathematical Foundations of Constructive Solid Geometry: General Topology of closed Regular Sets, Automata, Languages and Programming, University of Rochester, Rochester, New York, Production Automation Project, March 78.
14. Requiha, A.G., Representation for Rigid Solids : Theory, Methods, and Systems, Production Automation Project, The University of Rochester, New York, Computer Surveys, volume 12, number 4, 1980
15. Scott, D.S., Outline of a mathematical theory of computation, 4th Annual Princeton Conference on Information Sciences and Systems, pages 169-176, 1970.
16. Weihrauch, K., A Simple Introduction to Computable Analysis Informatik Berichte 171-2/1995- FernUniversität, Gesamthochschule in Hagen, Germany, 1995.