ISSN 2186-7437

# NII Shonan Meeting Report

No. 2013-13

# Implicit Computational Complexity and Applications: Resource Control, Security, Real-Number Computation

Akitoshi Kawamura
Jean-Yves Marion
David Nowak

November 4–7, 2013

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

# Implicit Computational Complexity and Applications: Resource Control, Security, Real-Number Computation

Organizers:
Akitoshi Kawamura (University of Tokyo, Japan)
Jean-Yves Marion (Lorraine University, France)
David Nowak (JFLI, CNRS & University of Tokyo, France & Japan)

November 4–7, 2013

Computational complexity theory aims at classifying computational problems according to their inherent difficulty. The standard way to achieve this classification consists in formalizing a precise execution model (e.g., a Turing machine) and posing explicit bounds on time and memory resources. On the other hand, Implicit Computational Complexity (ICC) aims at studying computational complexity without referring to external measuring conditions or a particular machine model, but only by considering language restrictions or logical/computational principles implying complexity properties. The area of ICC has grown out from several proposals to use logic and formal methods to provide languages for complexity-bounded computation (e.g., polynomial time, logarithmic space computation). ICC methods include, among others, linear logic, typed programming language, second order logic, term ordering. The last decades have seen the development of logical formalisms that characterize functions computable in various complexity classes (polynomial or elementary in time, logarithmic in space).

The goal of the proposed meeting is to explore foundational as well as practical interconnections between formal logic and computational complexity, such as it is done in ICC. The main outcome of this meeting will be to trigger new interactions and enrich the various approaches. In particular, and aside from traditional ICC approaches, we would like to focus on computation involving real numbers and topological spaces, thereby providing a deeper understanding of computational complexity in non-discrete realms of mathematics. By bringing together experts in implicit complexity and in complexity in analysis, we will promote new interaction between the two fields. People in those two fields are currently working separately, but there is enough common ground between them to make it worth having those two communities talking and working together. The meeting would also foster discussions about applications, i.e., the design of methods based on ICC and suitable for static verification of program resource consumption and of security.

Research topics for discussion on the various aspects described above would include, among others, the following topics:

- types for controlling complexity
- logical systems for computational complexity
- linear logic
- semantics of complexity-bounded computation
- rewriting and termination orderings
- interpretation-based methods for implicit complexity
- programming languages for complexity-bounded computation
- application of implicit complexity to the analysis of resource consumption
- application of implicit complexity to security
- complexity over reals and non-discrete spaces
- type-two complexity
- resource bounds in computable analysis and algorithmic randomness
- analog and continuous-time computation
- theory and implementation of efficient validated numerical algorithms

We briefly describe below two topics that we would like to promote. Although they originate from two different communities, they share the same interest in implicit characterization of complexity and they both have applications to formal validation of computer systems.

Very recently ICC methods have been applied to security methods and conversely security methods have been used as a new approach in ICC context. In the context of security proofs, the computational power of adversaries has to be limited so that their potential attacks are feasible. An adversary with unlimited computational power could indeed break most cryptographic schemes (e.g., RSA by efficiently factoring large integers). It is usual to rely on Cobhams thesis identifying feasibility with computability in polynomial time. Hence the particular interest in the class of functions computable in polynomial time and its implicit characterization with a programming language that can be used to construct adversaries. Conversely, type systems to control the information flow, which are traditionally used for certifying security policies like confidentiality or integrity, are related to the notion of data stratification. As a result, type systems for imperative programming languages have been developed to control resource consumption.

The other proposed focus is on computation over the reals. Computable analysis, the study of abilities and limitations of digital computers applied to problems in mathematical analysis, has originally evolved from computability theory, but there is increasing interest in computational complexity with bounded time and space. The goals here are to analyze the computational costs of algorithms for problems involving real numbers and to explore the principles and structures of computational complexity in this context, providing a foundation of validated numerical methods for problems arising in physical sciences and engineering. Broader perspectives in computational complexity, including those from implicit complexity theory, have high potential to help here, as can be already seen, for example, in recent studies of computational power of dynamical systems and analog computers, or in the application of type-two complexity theory to time-bounded computable analysis.

# 1 List of participants

1. Roberto Amadio, Department of Computer Science, University Paris Diderot (Paris 7)
2. Patrick Baillot, LIP, CNRS & ENS Lyon
3. Guillaume Bonfante, LORIA & University of Lorraine
4. Olivier Bournez, LIX, École Polytechnique
5. Ugo Dal Lago, Department of Computer Science and Engineering, University of Bologna
6. Hugo Férée, LORIA & University of Lorraine
7. Marco Gaboardi, School of Computing, University of Dundee
8. Nao Hirokawa, School of Information Science, Japan Advanced Institute of Science and Technology
9. Bruce Kapron, Department of Computer Science, University of Victoria
10. Akitoshi Kawamura, Department of Computer Science, University of Tokyo
11. Jean-Yves Marion, LORIA & University of Lorraine
12. Damiano Mazza, LIPN, University of Paris North (Paris 13)
13. Paul-André Melliès, PPS, CNRS & University Paris Diderot (Paris 7)
14. Georg Moser, Institute of Computer Science, University of Innsbruck
15. Jean-Yves Moyen, LIPN, University of Paris North (Paris 13)
16. David Nowak, LIFL, CNRS & Lille 1 University
17. Isabel Oitavem, CMAF-UL & FCT-UNL
18. Carsten Rösnick, Department of Mathematics, Darmstadt University of Technology
19. Jim Royer, Syracuse University
20. Ulrich Schöpp, Institute of Computer Science, Ludwig Maximilian University of Munich
21. Dieter Spreen, Department of Mathematics, University of Siegen & Department of Decision Sciences, University of South Africa, Pretoria
22. Klaus Weihrauch, Faculty of Mathematics and Computer Science, University of Hagen
23. Ning Zhong, Department of Mathematical Sciences, University of Cincinnati
24. Martin Ziegler, Department of Mathematics, Darmstadt University of Technology & JSPS BRIDGE program

## 2 Program

**November 3rd (Sunday)**

15:00–19:00 Hotel check-in (early check-in from 12:00 is possible)

19:00–21:00 Welcome Banquet

**November 4th (Monday)**

07:30–09:00 Breakfast (Cafetaria Oak)

09:00–09:10 Introduction of NII Shonan Meeting

09:10–12:00 Session 1:

Patrick Baillot
*An introduction to light logics, or Implicit complexity by taming the duplication (Introductory talk on ICC)*

Jean-Yves Moyen
*Interpretation methods in ICC (Introductory talk on ICC)*

12:00–13:30 Lunch (Cafetaria Oak)

13:30–14:00 Group photo shooting

14:00–18:00 Session 2:

Martin Ziegler
*Real complexity theory: a numerical view on "$\mathcal{P}$ vs. $\mathcal{NP}$" (Introductory talk on complexity in analysis)*

Akitoshi Kawamura
*Type-two complexity classes in computable analysis (Introductory talk on complexity in analysis)*

Carsten Rösnick
*About representations of and operators on subsets of* $\mathbb{R}^d$

Hugo Férée
*Higher order complexity and application in computable analysis*

18:00–19:30 Dinner (Cafetaria Oak)

**November 5th (Tuesday)**

07:30–09:00 Breakfast (Cafetaria Oak)

09:00–12:20 Session 3

Roberto Amadio
*Certifying and reasoning about cost annotations of functional programs*

Olivier Bournez
*On the complexity of solving ordinary differential equations. Towards a complexity theory for the General Purpose Analog Computer of Claude Shannon*

Isabel Oitavem
*Recursion schemes for P, NP and Pspace*

Klaus Weihrauch
*Products of effective topological spaces and a uniformly computable Tychonoff Theorem*

12:20–13:30 Lunch (Cafetaria Oak)

13:30–18:30 Session 4

Patrick Baillot
*Characterizing polynomial and exponential complexity classes in elementary lambda-calculus*

Georg Moser
*Amortised resource analysis and typed term rewriting*

Jim Royer
*Ramified structural recursion and corecursion*

Ulrich Schöpp
*Towards a primitive logic for computation by interaction*

Jean-Yves Moyen
*Expressive power and algorithms*

Bruce Kapron
*Computational soundness of symbolic security and implicit complexity*

18:30–19:30 Dinner (Cafetaria Oak)

### November 6th (Wednesday)

07:30–09:00 Breakfast (Cafetaria Oak)

09:00–12:30 Session 5

Ugo Dal Lago
*Infinitary lambda calculi from a linear viewpoint*

Marco Gaboardi
*Differential privacy: a type-based approach*

Nao Hirokawa
*Leftmost outermost and innermost strategies*

Dieter Spreen
*The continuity problem, once again*

Martin Ziegler
*The computational complexity of Laplace's equation*

12:30–13:30 Lunch (Cafetaria Oak)

13:30–19:00 Excursion (Kamakura)

18:00–20:00 Main banquet

### November 7th (Thursday)

07:00–10:00 Check-out

07:30–09:00 Breakfast (Cafetaria Oak)

09:00–12:30 Session 6

Damiano Mazza
*Infinitary lambda calculi and computational complexity*

Guillaume Bonfante
*Characterizing $NC^k$: from words to trees and back to words*

Paul-Andr Melliès
*Duplication and sharing in tensorial logic*

David Nowak
*Formal Security Proofs with ICC*

∗ Jean-Yves Marion
*Evolving graph structures, non-interference, and complexity*

12:30–13:30 Lunch (Cafetaria Oak)

## 3 Overview of talks

### Certifying and reasoning about cost annotations of functional programs

Roberto Amadio, Department of Computer Science, University Paris Diderot (Paris 7) (joint work with Yann Régis-Gianas)

### An introduction to light logics, or Implicit complexity by taming the duplication (Introductory talk on ICC)

Patrick Baillot, LIP, CNRS & ENS Lyon

In this talk we give a gentle introduction to the linear logic approach to implicit computational complexity. For that we first recall the proofs-as-programs reading of logic, with the system F of types for lambda-calculus. We then show how different variants of linear logic, ELL and LLL, can be seen as refinement of system F implying some complexity properties on typed programs.

### Characterizing polynomial and exponential complexity classes in elementary lambda-calculus

Patrick Baillot, LIP, CNRS & ENS Lyon

Elementary linear logic is a simple subsystem of linear logic that had been introduced to characterize elementary complexity, that is to say computation in time bounded by a tower of exponentials of fixed height. More recently it has been shown that the same logic, extended with type fixpoints, can actually be used to capture the complexity classes of problems P and $k$-EXPTIME, for $k \geq 1$. However these results relied on the use of proof-nets and on a delicate analysis of their normalisation procedure.

In the present work we retake such an investigation in the simpler setting of lambda-calculus. We define for that an elementary lambda-calculus, which is typable in elementary logic. We prove then that the hierarchy of complexity classes $k$-EXPTIME is characterized by a hierarchy of types, and extend this result from the classes of problems to the corresponding classes of functions.

### Characterizing $\mathrm{NC}^k$: from words to trees and back to words

Guillaume Bonfante, LORIA & University of Lorraine (joint work with Reinhard Kahle, Jean-Yves Marion and Isabel Oitavem)

In this talk, I come back to an analysis due to Leivant to describe in an algebraic way functions computable in $\mathrm{NC}^k$, that is by boolean circuits of depth $O(\log(n)^k)$ for $k > 0$. My talk will give two solutions to the problem, using two different data-structures: trees and words.

## On the complexity of solving ordinary differential equations: towards a complexity theory for the General Purpose Analog Computer of Claude Shannon

Olivier Bournez, LIX, École Polytechnique

## Infinitary lambda calculi from a linear viewpoint

Ugo Dal Lago, Department of Computer Science and Engineering, University of Bologna

## Higher order complexity and application in computable analysis

Hugo Férée, LORIA & University of Lorraine

The development of type two computability and complexity has allowed to define such notions over real numbers and other general spaces. It comes out that some spaces (for example the continuous functions over a non sigma-compact space) cannot be represented over the Baire space without defining a non intuitive notion of complexity. This motivated us to define a meaningful and robust notion of complexity at higher types based on game semantics, which can find many applications, not only in computable analysis.

## Differential privacy: a type-based approach

Marco Gaboardi, School of Computing, University of Dundee

Differential Privacy is becoming a standard approach in data privacy: it offers ways to answer queries about sensitive information while providing strong, provable privacy guarantees, ensuring that the presence or absence of a single individual in the database has a negligible statistical effect on the query's result. Many specific queries have been shown to be differentially private, but manually checking that a given query is differentially private can be both tedious and rather subtle. Moreover, this process becomes unfeasible when large programs are considered.

In this tutorial I will introduce the basics of differential privacy and some of the fundamental mechanisms for building differentially private programs. Additionally, I will present a type-based approach developed to help a programmer to certify his programs differentially private.

## Leftmost outermost and innermost strategies

Nao Hirokawa, School of Information Science, Japan Advanced Institute of Science and Technology

It is well-known that the leftmost outermost strategy is normalising for left-normal orthogonal rewrite systems. In this talk we give its simple proof.

## Computational soundness of symbolic security and implicit complexity

Bruce Kapron, Department of Computer Science, University of Victoria

Abadi and Rogaway initiated the study of computationally sound encryption logics in their landmark 2000 paper. Since then, their work has been developed and extended in a number of directions. The paradigmatic proof of computational soundness follows a reduction-based approach. We will present an alternate approach, based on interpretations in more general logics which axiomatise basic computational notions required for the formulation of cryptographic security.

## Type-two complexity classes in computable analysis (Introductory talk on complexity in analysis)

Akitoshi Kawamura, Department of Computer Science, University of Tokyo

A brief introduction to complexity in analysis viewed as application of type-two complexity classes.

## Evolving graph structures, non-interference, and complexity

Jean-Yves Marion, LORIA & University of Lorraine (joint work with Daniel Leivant)

## Infinitary lambda calculi and computational complexity

Damiano Mazza, LIPN, University of Paris North (Paris 13)

Recently, we introduced an infinitary lambda-calculus which is affine, in the sense that no duplication is possible (each variable occurs at most once). In such a calculus, the computational power given by duplication is recovered thanks to its infinitary character: in some sense, potential infinity is replaced by actual infinity. We will discuss how infinitary affine lambda-calculi may be used to recast some ideas, originated in linear logic, for defining calculi in which normalization is intrinsically time-bounded.

## Duplication and sharing in tensorial logic

Paul-André Melliès, PPS, CNRS & University Paris Diderot (Paris 7)

In this talk, I will explain how to characterize the free dialogue category with sums as a category of dialogue games and innocent strategies. I will also briefly indicate how to integrate duplication in the picture, in order to get a perfect correspondence between the proofs of tensorial logic with replication and a refined notion of dialogue game with backtracking.

## Amortised resource analysis and typed term rewriting

Georg Moser, Institute of Computer Science, University of Innsbruck (joint work with Martin Hofmann)

In this talk we present ongoing work on the connection between amortised cost analysis and polynomial interpretations. In particular we show that well-typedness of a constructor rewrite systems implies the existence of a (typed) polynomial interpretations orienting the rewrite system.

## Interpretation methods in ICC (Introductory talk on ICC)

Jean-Yves Moyen, LIPN, University of Paris North (Paris 13)

## Expressive power and algorithms

Jean-Yves Moyen, LIPN, University of Paris North (Paris 13) (joint work with Guillaume Bonfante and Pierre Boudes)

## Formal Security Proofs with ICC

David Nowak, LIFL, CNRS & Lille 1 University (joint work with Sylvain Heraud and Yu Zhang)

In this talk, we show how implicit computational complexity can be used in order to increase confidence in formal security proofs in cryptography.

## Recursion schemes for P, NP and Pspace

Isabel Oitavem, CMAF-UL & FCT-UNL

P, NP and Pspace are well-known classes of computational complexity that can be described following different approaches. Here we describe them in a machine independent manner, using recursion schemes, which turn the known inclusions P $\subseteq$ NP $\subseteq$ Pspace obvious. This work contributes to a better understanding of the involved classes, but no separation result is foreseen.

Recursion-theoretic approaches lead to classes of functions instead of predicates (or boolean functions). Therefore, instead of P and Pspace we reach the classes FPtime and FPspace. As a class of functions corresponding to NP we choose FPtime $\cup$ NP, and we adopt the notation FNPtime.

Our strategy is, as always in recursion-theoretic contexts, to start with a set of initial functions – which should be basic from the complexity point of view – and to close it under composition and recursion schemes. The recursion schemes can be bounded or unbounded depending on the chosen approach. In the first case we consider the Cobhams characterization of FPtime [2], in the second case we consider the Bellantoni-Cook characterization of FPtime [1]. In both cases we work over W, instead of N, where W is interpreted over the set of 0-1 words.

We look to these three classes of complexity – FPtime, FNPtime and FPspace – as resulting from three different models of computation – deterministic, non-deterministic and alternating Turing machines – and imposing the same resource constraint (polynomial time). Thus the adopted recursion schemes should somehow reflect the "increasing" computational power of the computation model. For FNPtime, besides the "calibration" of the recursion schemes, we have an additional problem since one is dealing with a class which is not closed under composition (because NP is not closed under negation). This work is based on [3] and [4].

[1] S. Bellantoni and S. Cook, A new recursion-theoretic characterization of Polytime functions, Computational Complexity, vol. 2 (1992), pp. 97–110.

[2] A. Cobham, The intrinsic computational difficulty of functions, Proc. of the 1964 International Congress for Logic, Methodology, and the Philosophy of Science, ed. Y. Bar-Hillel, North Holland, Amsterdam (1965), pp. 24–30.

[3] I. Oitavem, Characterizing Pspace with pointers, Mathematical Logic Quarterly, vol. 54 (2008), no. 3, pp. 317–323.

[4] A recursion-theoretic approach to NP, Annals of Pure and Applied Logic, vol. 162 (2011), no. 8, pp. 661–666.

## About representations of and operators on subsets of $\mathbb{R}^d$

Carsten Rösnick, Department of Mathematics, Darmstadt University of Technology

There are many representations (notion from TTE) of sets that all have been compared in the past, but mostly regarding their computable equivalence. By giving a few proof sketches we show some of them to be (parameterized) polynomial-time equivalent. Using this insight, we then present examples of operators on sets (like Intersection or Projection) and functions+sets (like FunctionInversion), plus their respective complexity bounds.

## Ramified structural recursion and corecursion

Jim Royer, Syracuse University (joint work with Norman Danner)

We investigate the notion of feasible computability over inductive structures (henceforth, data) and coinductive structures (henceforth, codata). Data includes: natural numbers, strings of characters, (finite) lists, (finite) trees, etc. Codata includes: various sorts of potentially infinite lists (streams) and potentially infinite trees. We also investigate simple typed programming formalisms that attempt to capture feasible computability over data and codata.

Pinning down "feasible computability"1q over data and codata and capturing it in a programming formalisms turn out to be a subtle problem for which there is not yet a final answer. The present work focuses on bettering our understanding these problems and crafting tools to deal with them.

We start by introducing some simple classical systems for computing over data (via structural recursions) and codata (via structural corecursions) whose computational power goes well beyond the feasible. We rein in these systems by imposing a simple, two level ramification of their types (based on Bellantoni and

Cook's normal/safe distinction). The resulting systems are nice and simple and a little surprising in what they compute and what they seem to fail to compute, especially in the codata realm.

The talk will be a tour through these various systems, focusing on motivations, definitions, examples, counterexamples, and open problems that our work reveals.

## Towards a primitive logic for computation by interaction

Ulrich Schöpp, Institute of Computer Science, Ludwig Maximilian University of Munich

Computation by Interaction is an approach of viewing computation as a dialogue between a number of entities. In the context of implicit computational complexity this approach has been applied to obtain characterisations of (N)LOGSPACE; there are many other applications in other contexts, such as a recent approach to seamless distributed computation by Fredriksson and Ghica, for example. After giving a general overview of Computation for Interaction, I will talk about ongoing work of capturing essential logical principles of Computation by Interaction in terms of a simple fragment of linear logic.

## The continuity problem, once again

Dieter Spreen, Department of Mathematics, University of Siegen & Department of Decision Sciences, University of South Africa, Pretoria

Let $(T, \tau)$ be an effectively given topological space. As usual in Russian-style constructive mathematics, only spaces consisting of computable elements are considered. In order to compute with elements, these are coded by natural numbers. Then, on the one hand, $T$ is an object of the category of numbered structures, which has effective maps as morphisms. Here, a map is *effective* if it is tracked by a computable function on the code. On the other hand, $T$ is an object in the category of effectively given topological spaces, which has effectively continuous maps as morphisms. These are maps for which the preimage of any basic open set is an effective union of basic open sets the index of which uniformly depends on the index of the given basic open set. The *continuity problem* is the question whether effective maps are effectively continuous. For several important examples like the space of all partial recursive functions, or the computable reals, the answer is yes. In the general case it was shown by the speaker that a map between two effectively given spaces is effectively continuous if, and only if, it is effective and has a witness for non-inclusion.

In this talk it is shown that the proof of the important direction can be split into two parts: First it is shown that every effective function is effectively sequentially continuous, i.e., transforms computably convergent computable sequences into the same kind of sequences. Then, by following the classical proof and using witnesses for non-inclusion, it is shown that every such map is effectively continuous.

## Products of effective topological spaces and a uniformly computable Tychonoff Theorem

Klaus Weihrauch, Faculty of Mathematics and Computer Science, University of Hagen (joint work with Robert Rettinger)

This article is a fundamental study in computable analysis. In the framework of Type-2 effectivity we investigate computability aspects on finite and infinite products of effective topological spaces. For obtaining uniform results we introduce natural multi-representations of the class of all effective topological spaces, of their points, of their subsets and of their compact subsets. We formulate a meta-theorem by which there is a function computable uniformly in the spaces if there is a computable function for every computable space. We study computability of the product operations on points, on arbitrary subsets and on compact subsets and computability of their inverses. For the case of compact sets the results are uniformly computable versions of Tychonoff's Theorem (stating that the every cartesian product of compact spaces is compact) for both, the cover multi-representation and the "minimal cover" multi-representation.

## Real complexity theory: a numerical view on "$\mathcal{P}$ vs. $\mathcal{NP}$" (Introductory talk on complexity in analysis)

Martin Ziegler, Department of Mathematics, Darmstadt University of Technology & JSPS BRIDGE program

We promote Real Complexity Theory as a bridge between numerical practice (actual implementations solving 'many' instances efficiently heuristically) and Computability in Analysis (non-/existence of algorithms guaranteed to solve all instances from a specified class – eventually): For a provably correct algorithm, upper bounds on its asymptotic time or memory consumption permit to assess its potential for practical competitiveness; and lower bounds establish its optimality. The consistent semantics underlying "*feasible* real random access machines" (Brattka&Hertling'98) is closed under composition: a significant structural advantage and key to the modular development of reliable real libraries like `iRRAM`. By comparing representations, TTE allows to identify the information (continuous approximations enriched with discrete advice) sufficient and necessary for rendering a real problem uniformly computable – which yields a canonical `C++` class interface declaration for actual implementations.

Now the time to evaluate a computable real function $x \mapsto f(x)$ up to error $1/2^n$ generally depends on both $n$ and $k := \lceil |x| \rceil$, leading to a parameterized complexity theory. More generally the complexity of a functional $\Lambda : C[0;1] \ni f \mapsto \Lambda(f) \in \mathbf{R}$ depends on $n$ and on the given modulus of continuity $\mu : \mathbf{N} \to \mathbf{N}$ of $f$: a second-order parameter in the sense of Kapron&Cook (1996).

Concerning operators, Maximum has been shown to map polytime (and even smooth such) $f : [0;1] \to [0;1]$ to functions that are $\mathcal{NP}$–'complete' in a sense formalized by Kawamura&Cook (STOC2010); and Riemann integration even corresponds to $\#\mathcal{P}$.

## The computational complexity of Laplace's equation

Martin Ziegler, Department of Mathematics, Darmstadt University of Technology & JSPS BRIDGE program (joint work with Akitoshi Kawamura and Florian Steinberg)

The last years have seen an increasing interest in classifying (existence claims in) classical mathematical theorems according to their strength: in terms of the underlying logical systems sufficient and necessary for their proof as well as in terms of their computational power, that is, (e.g. WEIHRAUCH-)degrees of computability. We pursue the latter under the refined view of asymptotic computational worst-case complexity.

H. FRIEDMAN and KER-I KO (1982) have shown the indefinite integral operator $C[0;1] \ni f \mapsto \big(x \mapsto \int_0^x f(t)\,dt\big)$ to map polytime computable (even smooth) $f : [0;1] \to [0;1]$ to $\#\mathcal{P}$–'complete' functions. And the solution $u$ to the ordinary differential equation $u'(t) = f\big(t, u(t)\big)$ with polynomial-time computable $C^1$ right-hand side $f : [0;1] \times [-1;1] \to [-1;1]$ and initial condition $u(0) = 0$ can even be as complex as $\mathcal{PSPACE}$ (KAWAMURA 2010ff). Concerning single reals, $\|f\|_1$ and $\int_0^1 f(t)\,dt$ and $u(1)$ can be as hard as the unary/tally complexity classes $\mathcal{NP}_1 \subseteq \mathcal{NP}$ and $\#\mathcal{P}_1 \subseteq \#\mathcal{P}$ and $\mathcal{PSPACE}_1 \subseteq \mathcal{PSPACE}$, respectively.

We explore analogous questions for simple elliptic partial differential equations with computable classical solutions (as opposed to the hyperbolic Wave equation: POUR-EL&RICHARDS 1981).

(1) $\Delta u = 0$ on $\Omega$, $\quad u = g$ on $\partial\Omega$ (2) $\Delta u = f$ on $\Omega$, $\quad u = 0$ on $\partial\Omega$

for fixed open domains $\Omega \subseteq [-1;1]^d$ and continuous/smooth $f : \overline{\Omega} \to [0;1]$ and $g : \partial\Omega \to [0;1]$.