Math. Struct. in Comp. Science (2017), *vol.* 27, *pp.* 1437–1465. © Cambridge University Press 2016
doi:10.1017/S096012951600013X First published online 28 July 2016

On the computational complexity of the Dirichlet Problem for Poisson's Equation

AKITOSHI KAWAMURA[†], FLORIAN STEINBERG[†,‡] and MARTIN ZIEGLER[‡]

[†]*Graduate School of Arts and Sciences, University of Tokyo, Tokyo, Japan*
[‡]*Department of Mathematics, TU Darmstadt, Darmstadt, Germany*

Received 31 March 2014; revised 10 April 2015

The last years have seen an increasing interest in classifying (existence claims in) classical mathematical theorems according to their strength. We pursue this goal from the refined perspective of computational complexity. Specifically, we establish that rigorously solving the Dirichlet Problem for Poisson's Equation is in a precise sense 'complete' for the complexity class $\#\mathcal{P}$ and thus as hard or easy as parametric Riemann integration (Friedman 1984; Ko 1991. *Complexity Theory of Real Functions*).

1. Introduction

Friedman (1984), Ko and Friedman (1982) and Ko (1982) have proven that, for a polynomial-time computable (and even for a smooth, i.e., infinitely often differentiable) function $h : [0;1] \to \mathbb{R}$ the parametric maximum $[0;1] \ni x \mapsto \max\{h(t) \mid 0 \leqslant t \leqslant x\}$ can have complexity exhausting $\mathcal{NP}$. Put differently, any algorithm for computing maxima of (smooth) real functions up to guaranteed absolute error 2^{-n} within time exponential in n is essentially optimal! This result has set off a series of further characterizations of discrete complexity classes in terms of numerical problems (Braverman 2005; Kawamura 2010; Ko 1991, 1990, 1992, 1998; Ko and Yu 2008; Rösnick 2013). In particular, computing the Riemann integral has been shown (Friedman 1984, Theorem 5.33) to correspond to the even larger complexity class $\#\mathcal{P}$ in the following precise sense:

Fact 1.1. The following are equivalent:

i. $\mathcal{FP} = \#\mathcal{P}$.
ii. For every polynomial-time computable $h : [0;1] \to \mathbb{R}$ the function

$$\int h : [0;1] \to \mathbb{R}, \quad x \mapsto \int_0^x h(t)\,dt$$

is again polynomial-time computable.
iii. For every smooth polytime $h : [0;1] \to \mathbb{R}$ with support $\mathrm{supp}(f) \subseteq [1/4;3/4]$, $\int h$ is again polytime.

Recall that $\mathcal{FP}$ is the class of mappings $\varphi : \{0,1\}^* \to \mathbb{N} = \{0,1,2,\ldots\}$ computable by a deterministic Turing machine within time polynomial in the binary length of the input; $\mathcal{NP}$ is the class of decision problems accepted by a non-deterministic Turing machine within polynomial time; and $\#\mathcal{P}$ is the class of mappings φ counting the number of accepting computations of a non-deterministic polynomial-time Turing machine (cf. Parberry and Schnitger (1988)). We also report that definite integrals like $\int_0^1 h(t)\,dt$ correspond to the complexity class $\#\mathcal{P}_1$ (Ko 1991, Theorem 5.32) and are thus likely easier than indefinite ones.

Recall that the Dirichlet Problem for Poisson's Equation on a domain $\Omega \subseteq \mathbb{R}^d$ is to find, given functions $f : \Omega \to \mathbb{R}$ and $g : \partial\Omega \to \mathbb{R}$, a function $u : \overline{\Omega} \to \mathbb{R}$ satisfying

$$\Delta u = f \text{ in } \Omega, \quad u|_{\partial\Omega} = g. \tag{1}$$

The main result of the present work extends Fact 1.1 with a new numerical characterization of the discrete complexity class $\#\mathcal{P}$. To this end, fix a dimension $d \geqslant 1$ and let $B^d := \{ \times \in \mathbb{R}^d \mid \|\boldsymbol{x}\| < 1 \}$ denote the open Euclidean unit ball.

Theorem 1.2. Any of the items in Fact 1.1 above is furthermore equivalent to:

iv. For every choice of polynomial-time computable functions $f : B^d \to \mathbb{R}$ and $g : \partial B^d \to \mathbb{R}$, the solution $u : \overline{B}^d \to \mathbb{R}$ to the Dirichlet Problem for Poisson's Equation (1) on B^d is again computable in polynomial time.

v. For every choice of smooth polynomial-time computable functions $f : B^d \to \mathbb{R}$ and $g : \partial B^d \to \mathbb{R}$, the solution $u : \overline{B}^d \to \mathbb{R}$ to the Dirichlet Problem for Poisson's Equation (1) on B^d is again computable in polynomial time.

The easy one-dimensional case of this theorem is treated separately in Section 1.1. Our presentation supposes familiarity with discrete computability and classical analysis. In Section 2, we recall basic notions and facts from Computable Analysis and Real Complexity Theory. Section 3 reports on some known analytic properties of partial differential equations; and establishes in Lemma 3.8 that, for polynomial-time computable f, the solution u is indeed classical (i.e. twice continuously differentiable) rather than just a weak one. A variation of Fact 1.1 which we later build upon is proven in Section 4. Section 5 presents a rigorous algorithm for solving the Dirichlet problem for Poisson's Equation on the unit ball within $\#\mathcal{P}$: exploiting linearity, separately for the case $f = 0$ and for the case $g = 0$. Specifically, Section 5.1 describes, analyses and proves correctness of our algorithm for the general Dirichlet problem for Laplace's Equation; and Section 5.2 treats the homogeneous Dirichlet problem for Poisson's Equation. Section 6 finally establishes the optimality of said algorithm by showing that a polynomial-time solution to Equation (1) on the unit ball implies $\#\mathcal{P} = \mathcal{FP}$.

1.1. *One-dimensional case*

Concerning $d = 1$ as a motivation and guide towards the following considerations, Equation (1) boils down to

$$\ddot{u} = f, \qquad u(-1) = g_-, \quad u(1) = g_+$$

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

where we used the notation $\ddot{u}$ for the second derivative with respect to the only variable. The solution can be specified as

$$u(x) = \int_{-1}^{x} (x-t) \cdot f(t)\,dt + Cx + C' = \int_{-1}^{x}\int_{-1}^{t} f(s)\,ds\,dt + Cx + C',$$
$$\text{where } C := (g_+ - g_- - E)/2, \quad C' := (g_+ + g_- - E)/2, \tag{2}$$
$$\text{and } E := \int_{-1}^{1} (1-t) \cdot f(t)\,dt = \int_{-1}^{1}\int_{-1}^{t} f(s)\,ds\,dt.$$

So for polynomial-time f and g_-, g_+, the condition of Fact 1.1 (ii) yields polynomial-time computability of u. Conversely, assume that the solution u is polynomial-time computable whenever f and g_+ and g_- are. Then Fact 1.1 (iii) will hold: For let $h : [0;1] \to \mathbb{R}$ be a smooth function with $\mathrm{supp}(h) \subseteq [1/4;3/4]$. Extend h to an odd, smooth polynomial-time computable function on $[-1;1]$. It is easy to see that the derivative $f := \dot{h}$ of any polynomial-time computable $h \in C^2[-1;1]$ is again polynomial-time computable (cf. the slightly more general Proposition 2.3c). In Equation (2), we have $E = \int_{-1}^{1} h(t)\,dt = 0$ due to symmetry. If we choose $g_+ = g_- = 0$ it also follows that $C = C' = 0$ and therefore $u(x) = \int_{-1}^{x} h(t)\,dt$. So the polynomial-time computability of $\int_0^x h(t)\,dt = u(x) - u(0)$ follows from the polynomial-time computability of u.

1.2. *Notations*

We close the introduction with some basic notational conventions used throughout the rest of the paper.

1.2.1. *Notions from computability theory:* For $n \in \mathbb{N}$ set

$$\mathbb{D}_n := \left\{ \frac{k}{2^n} \,\middle|\, k \in \mathbb{Z} \right\} \quad \text{and} \quad \mathbb{D} := \bigcup_{n \in \mathbb{N}} \mathbb{D}_n .$$

We call the elements of $\mathbb{D}$ *dyadics*. Real numbers that can be written in this form are called *dyadic numbers*. (Note that, as opposed to dyadic numbers, dyadics have a fixed denominator and thus disallow cancelling: a minor restriction that will later simplify their coding and complexity estimates.) An interval is dyadic if its end points are dyadic; a point in $\mathbb{R}^d$ is dyadic if its components are dyadic, and a square if its vertices are dyadic.

Call an element of $\Omega \subseteq \mathbb{R}^d$ a *dyadic point* of Ω if it lies within the interior and is a dyadic point of $\mathbb{R}^d$. An element of ∂B is a *spherically dyadic point* if it has dyadic spherical coordinates. (These notions will be justified in Fact 2.7 below.)

Given $n \in \mathbb{N}$, write $[\![n]\!] \in \{0,1\}^*$ for its unique *binary encoding* without leading zeros; and $1^n \in \{1\}^*$ for its *unary encoding*. We denote the *binary length* of some element $\boldsymbol{a} \in \{0,1\}^*$ by $l(\boldsymbol{a})$. $l(n)$ is an abbreviation of $l([\![n]\!])$.

Let $\langle \cdot,\cdot \rangle : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ denote some polynomial-time computable pairing function with polynomial-time computable inverse, such that $l(\langle \boldsymbol{a},\boldsymbol{b} \rangle) \leqslant P(l(\boldsymbol{a}) + l(\boldsymbol{b}))$ for some polynomial P. (The reader might imagine interleaving both arguments with trailing makers, although the specifics do not matter here. . .) We use this map to define a binary encoding of tuples of integers, rational numbers, dyadics and other Cartesian products.

For example, dyadics from $\mathbb{D}_n$ are encoded by

$$\left[\!\!\left[\frac{k}{2^n}\right]\!\!\right] := \langle [\![k]\!], [\![2^n]\!] \rangle.$$

(Note that elements of $\mathbb{D}_n$ will always have the binary encoding of 2^n in their second component. We explicitly disallow cancelling in this case). Computability and complexity of continuous objects will then build on this encoding and its length. Writing $\phi :\subseteq X \to Y$ indicates that the function ϕ is partial, that is, is defined only on some possibly proper subset of X.

1.2.2. *Notions from analysis:* For a subset A of a topological space, denote its *closure* by $\overline{A}$, its *interior* by $\mathring{A}$ and its *boundary* by $\partial A = \overline{A} \setminus \mathring{A}$. $C(A)$ means the set of continuous functions from A to $\mathbb{R}$, equipped with the *supremum norm* $\| \cdot \|_\infty$.

If $A \subseteq \mathbb{R}^d$ is open, we write $C^m(A)$ for the m times continuously differentiable functions on A, and set $C^\infty(A) := \bigcap_{m \in \mathbb{N}} C^m(A)$. Call elements of $C^\infty(A)$ the smooth functions on A. A claim that $f : \overline{A} \to \mathbb{R}$ lies in one of those spaces means that the derivatives of the restriction of f to A exist and extend continuously to $\overline{A}$. The usual notations for partial derivatives: $D_i f$ is for $\frac{\partial f}{\partial x_i}$, $D_{ij} f$ is for $D_i(D_j f)$ and

$$Df := \begin{pmatrix} D_1 f \\ \vdots \\ D_d f \end{pmatrix}.$$

We will mostly consider the *Euclidean norm* on $\mathbb{R}^d$, written as $\| \cdot \|$. For $\boldsymbol{x} \in \mathbb{R}^d$ and $r > 0$, let $B_r^d(\boldsymbol{x}) := \{\boldsymbol{y} \in \mathbb{R}^d \mid \|\boldsymbol{y} - \boldsymbol{x}\| < r\}$ denote the d-dimensional open ball of radius r around x, and abbreviate $B_1^d(0)$ by B^d. If the dimension is clear from the context it will sometimes be omitted. In particular, ∂B^d is the $(d-1)$-dimensional unit sphere, and $\overline{B}^d$ the closed unit ball.

The Lebesgue measure on $\mathbb{R}^d$ is denoted by λ, and the spherical measure on ∂B by σ. In particular, $\lambda(B) = \frac{\pi^{d/2}}{\Gamma(d/2+1)}$ is the volume of the unit ball and $\sigma(\partial B) = d \cdot \lambda(B)$ the surface area of the unit sphere. While we write $d\sigma(\boldsymbol{y})$ for integration with respect to σ, Lebesgue integration may be shortened to $d\boldsymbol{y}$.

Many functions have their standard names: sin, cos, e^x, etc. We use lb for the *binary logarithm* and ln for the *natural logarithm*. For a continuous function h on the unit interval, the function

$$x \mapsto \int_0^x h(t)\, dt$$

is its *antiderivative*.

2. Recap on computability and complexity in analysis

Computability theory for real numbers has been devised together with its discrete, now so-called classical, counterpart (Turing 1936). It was then extended to functions (Grzegorczyk

1957), closed/open Euclidean subsets (Lacombe 1958), operators (Pour-El and Richards 1989) and other universes of continuum cardinality (Weihrauch 2000).

Numerical solutions to partial differential equations are a big topic, both in engineering and mathematics. However, rigorous algorithmic solutions, and even establishing cases of provable incomputability, lies within the core expertise of Recursive Analysis (Brattka and Yoshikawa 2006; Sun et al. 2015; Weihrauch and Zhong 2002, 2005, 2006, 2007). Concerning the refined view of computational complexity, investigations have recently moved to ordinary differential equations (Bournez et al. 2013; Kawamura 2010).

This section recalls some of the basic notions. Our presentation here is tailored and limited to those aspects employed towards the main result. For a more thorough introduction into this field, and for the proofs of the cited facts, the reader is invited to refer to a standard textbook like Ko (1991) or Weihrauch (2000, Section 7). Out of several common, equivalent ways of defining polynomial-time computability for single real numbers or vectors, we proceed from the following:

Definition 2.1.

a. A real vector $\boldsymbol{x} \in \mathbb{R}^d$ is *computable* if there exists a computable sequence $(\boldsymbol{q}_n)_{n\in\mathbb{N}}$ of vectors of dyadics $\boldsymbol{q}_n \in \mathbb{D}_n^d$, which converges towards $\boldsymbol{x}$ with 'binary' speed, i.e., such that $\|\boldsymbol{q}_n - \boldsymbol{x}\| \leqslant 2^{-n}$.
b. It is said to be *polynomial-time computable* if such a sequence exists computable within time polynomial in the value of n.

We refrain from formally defining computability and polynomial-time computability for real functions as the sequel will conveniently build on an equivalent characterization reported in Fact 2.7. For an intuitive conception, fix a reasonable encoding of dyadic sequences $(\boldsymbol{q}_n)_{n\in\mathbb{N}}$ as infinite binary strings. Then, according to the type-two theory of effectivity (TTE), computability of $f : \Omega \subseteq \mathbb{R}^d \to \mathbb{R}$ is defined by means of a machine that, whenever started with the input tape holding such an infinite binary sequence encoding some real vector $\boldsymbol{x} \in \Omega$, will write an encoding of $f(\boldsymbol{x})$ onto a dedicated one-way output tape. (The machine's behaviour on input sequences not encoding any $\boldsymbol{x} \in \Omega$ may be arbitrary...) Quantitative time complexity arises from the requirement that the machine prints the nth binary output digit within a number of steps bounded by some function $t(n)$, preferably a polynomial in n. Note that this running time bound may thus depend only on the output precision but not on the real argument: rendering for instance the reciprocal function $(0;1] \ni x \mapsto 1/x$ computable, but not within bounded time.

Example 2.2. For any dyadic sequence $(a_m)_{m\in\mathbb{N}}$ computable within time polynomial in m (in the discrete sense of receiving m as input in unary and producing numerator and denominator of a_m in binary) with radius of convergence $R = 1/\limsup_m \sqrt[m]{|a_m|} > 0$ and for any fixed $r \in (0;R)$, the associated power series function $[-r;r] \ni x \mapsto \sum_m a_m x^m$ is polynomial-time computable.

In particular, the usual entire functions sin, cos and e^x are polynomial-time computable when restricted to any bounded real interval.

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

More details about polynomial-time computability of analytic functions can for example be found in Müller (1995).

We will routinely use the following closure properties of the class of polynomial-time computable functions:

Proposition 2.3.

a. The sum and pointwise product of polynomial-time computable functions is again computable in polynomial time.
b. The composition of polynomial-time computable functions is again computable in polynomial time.
c. If $\Omega \subseteq \mathbb{R}^d$ is open and $f \in C^2(\overline{\Omega})$ is polynomial-time computable, then the partial derivatives $D_i f$ of f are polynomial-time computable again.

For a more thorough discussion of Item c refer to Ko (1991, Section 6). Also note that there exists a polynomial-time computable function $f \in C^1[0;1]$ whose derivative is not computable at all (Zhong 1998).

It is true, and not hard to prove, that computable functions are necessarily continuous: an observation sometimes called the 'Main Theorem' of Computable Analysis. In fact, for functions over well-behaved domains, both computability and complexity split into a discrete and a topological condition – based on the following quantitative refinement of uniform continuity.

Definition 2.4. For $\Omega \subseteq \mathbb{R}^d$, consider a function $f : \Omega \to \mathbb{R}$. A mapping $\mu : \mathbb{N} \to \mathbb{N}$ is called a *modulus of continuity for* f if for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$

$$\|\boldsymbol{x} - \boldsymbol{y}\| \leqslant 2^{-\mu(n)} \;\Rightarrow\; |f(\boldsymbol{x}) - f(\boldsymbol{y})| \leqslant 2^{-n}.$$

A modulus of continuity thus specifies sufficient conditions on the arguments' proximity (in binary) in order for the values to be close. Obviously, any function that admits a modulus of continuity must be uniformly continuous. Conversely, every continuous function on a compact space will have a modulus of continuity.

Example 2.5. For $\Omega \subseteq \mathbb{R}^d$, consider $f : \Omega \to \mathbb{R}$, and suppose the image of f is bounded. (This is for example the case whenever Ω is compact and f is continuous.) Then Lipschitz continuity means having a modulus of continuity of the form $\mu(n) = n + b$; and Hölder continuity corresponds to linear moduli of continuity $\mu(n) = an + b$.

More precisely, if f satisfies

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leqslant C\|\boldsymbol{x} - \boldsymbol{y}\|^\alpha, \tag{3}$$

then $\mu(n) := \frac{n+\mathrm{lb}(C)}{\alpha}$ will be a modulus of continuity. Conversely, for $\mu(n) = an + b$ a modulus of continuity of f and M a bound on the diameter of the image of f, considering the cases $2^{\mu(n+1)} \leqslant \|\boldsymbol{x} - \boldsymbol{y}\| < 2^{-\mu(n)}$ for some n and $2^{-\mu(0)} \leqslant \|\boldsymbol{x} - \boldsymbol{y}\|$ shows that

$$\alpha := \frac{1}{a} \quad \text{and} \quad C := \max\{2^{\alpha b} M, 2^{\alpha b + 1}\}$$

satisfy Equation (3).

For bounded-time computations, the aforementioned 'Main Theorem' refines from a qualitative to a quantitative topological requirement: Computing $f : \Omega \to \mathbb{R}$ in time $t(n)$ implies that $\mu(n) := t(n+1)+1$ is a modulus of continuity for f. The argument, referring to the informal notion introduced below Definition 2.1, roughly proceeds as follows: A machine computing f within $t(n)$ steps can, before producing an approximation to $f(\boldsymbol{x})$ up to error 2^{-n}, have read at most $m \leqslant t(n)$ approximations up to error 2^{-m} to the input $\boldsymbol{x}$ and therefore cannot distinguish from any $\boldsymbol{x}' \in \Omega$ with $\|\boldsymbol{x}-\boldsymbol{x}'\| \leqslant 2^{-m-1}$. Moreover output approximations up to error 2^{-0} produced within $t(0)$ steps can have binary length at most $t(0)$, yielding $2^{t(0)}+1$ as upper bound on f.

The Poisson problem involves continuous functions f on the open, not the closed, unit ball. However, by the above considerations, a polynomial-time computable function is bounded and uniformly continuous. It thus automatically extends continuously to the closure of the domain – also in time $t(n)$:

Proposition 2.6. Fix $\Omega \subseteq \mathbb{R}^d$ and suppose $f : \Omega \to \mathbb{R}$ is computable within bounded time $t(n)$. Then f admits a (unique) continuous extension to $\overline{\Omega}$ computable within the same time $t(n)$.

This can be proved by arguing that the same (TTE) machine computes a function on the closure of the domain.

This in mind, the next characterization of polynomial-time computability should not come as a surprise and will be used as definition of computability and complexity for real functions in the rest of the paper:

Fact 2.7. Let $f : \Omega \to \mathbb{R}$ be a function, where $\Omega \in \left\{\overline{B}^d, \partial B^d, [0;1]^d\right\}$. The following are true:

1. f is computable if and only if the following conditions are fulfilled:
   – There is a partial computable function $\phi :\subseteq \{0,1\}^* \to \mathbb{D}$ such that whenever $\boldsymbol{q}$ is a (spherical, in the case $\Omega = \partial B^d$) dyadic point of Ω and $n \in \mathbb{N}$ we have
   $$|f(\boldsymbol{q}) - \phi(\langle [\![\boldsymbol{q}]\!], 1^n\rangle)| \leqslant 2^{-n}.$$
   – f has a computable modulus of continuity.
2. f is polynomial-time computable if and only if the following conditions are fulfilled:
   – There is a polynomial-time computable function ϕ as above.
   – f has a polynomial modulus of continuity.

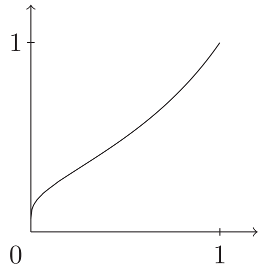
A proof for the case $\Omega = [0;1]^d$ can be found in Ko (1991, Section 2). Concerning Item 2, the partial function ϕ can in this case be made total by having it abort after exceeding the polynomial running time bound. Similarly, every polynomial modulus of continuity is polynomial-time computable as a unary function from $\{1\}^*$ to $\{1\}^*$, and vice versa.

Fact 2.7 justifies the notions of dyadic and spherical dyadic points we introduced. Indeed, in this case the different domains should be considered as manifolds (but a formal generalization to compact manifolds with bi-computable, respectively bi-polynomial-time computable charts is beyond our purpose).

Example 2.8.

a. The function

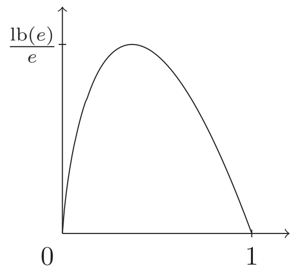
$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ \frac{1}{1-\operatorname{lb}(x)} & \text{if } x \neq 0 \end{cases}.$$



admits an exponential but no polynomial modulus of continuity. Therefore, it is not computable in polynomial time.

b. The function

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ -x\operatorname{lb}(x) & \text{if } x \neq 0 \end{cases}.$$



is polynomial-time computable: Since f is Hölder continuous, it has a polynomial modulus of continuity (cf. Example 2.5). Moreover, it is not hard to compute the binary logarithm of some non-zero dyadic number within time quadratic in the binary length of that number. Since multiplication is also possible in quadratic time, we get an algorithm for evaluating f on non-zero dyadic arguments.

Note that we required the dyadic points of a set to lie within the interior of that set, thus the algorithm needs not be defined if the input is 0 or 1. On the other hand, it is not difficult to make it also return valid approximations on these inputs as it can use the modulus of continuity and evaluation on an interior point close to the boundary point.

3. Partial differential equations

The classical theorems of Peano and Picard–Lindelöf assert local existence and uniqueness of solutions to systems of ordinary differential equations $\dot{\boldsymbol{u}}(t) = f\big(\boldsymbol{u}(t), t\big)$. They proceed from modest and natural hypotheses (Lipschitz-/continuity) on the right-hand side f under the reasonable notion of solution $\boldsymbol{u} \in C^1$. Concerning partial differential equations, however, the questions of existence and uniqueness become much more involved, already in the linear case. Specifically, for Poisson's Equation (1) to make sense, one needs $u \in C^2(\Omega) \cap C(\overline{\Omega})$. However, for the two-dimensional unit ball $\Omega = B^2$, Henrik Petrini specified a continuous right-hand side f provably admitting no C^2 (i.e., classical) solution u (Wienholtz et al. 2009, Satz 4.3.1). Indeed, the theory of partial differential equations

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

knows a rich variety of generalized notions of a solution: strong, weak, mild, in Hilbert, L^p, and (possibly weighted and fractional) Sobolev spaces, to name a few of them.

The present section recollects some basic background about Laplace's and Poisson's Equation. We adapt and improve known results on the regularity of their solutions that conveniently rule out Petrini's counter-example for polynomial-time computable f. In particular, the present work gets by only briefly touching weak solutions at the beginning of Section 3.2.

In the sequel fix a dimension $d > 1$ and leave away the usual superscript. Thus, B will denote the d-dimensional unit ball etc.

Recall that the Laplace operator Δ on an open set $\Omega \subseteq \mathbb{R}^d$ is given by

$$\Delta : C^2(\Omega) \to C(\Omega), \quad u \mapsto \sum_{i=1}^{d} D_{ii}u = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2}.$$

The Dirichlet Problem for Poisson's Equation on a domain $\Omega \subseteq \mathbb{R}^d$ is to find, given functions $f : \Omega \to \mathbb{R}$ and $g : \partial\Omega \to \mathbb{R}$, a function $u : \overline{\Omega} \to \mathbb{R}$ satisfying

$$\Delta u = f \text{ in } \Omega, \quad u|_{\partial\Omega} = g. \tag{4}$$

By linearity of the Laplacian, solving the Dirichlet Problem for Poisson's Equation can be divided into two parts: the case of Laplace's Equation, that is, setting $f = 0$ in Equation (4); and the homogeneous case of Poisson's Equation, that is, setting $g = 0$ in Equation (4). We will now investigate these problems separately in more detail for the case $\Omega = B$.

3.1. *Laplace's equation*

Consider the Dirichlet problem for Laplace's equation on the unit ball:

$$\Delta u = 0 \text{ in } B, \quad u|_{\partial B} = g \tag{L}$$

where $g : \partial B \to \mathbb{R}$ is some function on the boundary.

It is known that such a solution u exists and is unique whenever g is continuous. This renders the operator $g \mapsto u$ well defined on $C(\partial B)$. In fact an 'explicit' solution is given by the *Poisson integral*:

$$w : B \to \mathbb{R}, \quad \boldsymbol{x} \mapsto \int_{\partial B} K(\boldsymbol{x}, \boldsymbol{y}) \cdot g(\boldsymbol{y})\, d\sigma(\boldsymbol{y}), \tag{PI}$$

where

$$K(\boldsymbol{x}, \boldsymbol{y}) = \frac{1 - \|\boldsymbol{x}\|^2}{d \cdot \lambda(B) \cdot \|\boldsymbol{x} - \boldsymbol{y}\|^d} \tag{K}$$

is the *Poisson Kernel* (see Figure 1). Indeed, we have:

Fact 3.1. Whenever $g \in C(\partial B)$, the function w continuously extends to a solution $u \in C^2(B) \cup C(\overline{B})$ of Equation (L).

A standard proof can for example be found in Gilbarg and Trudinger (2001, Theorem 2.6). A refined analysis of this proof will lead to Lemma 3.4. Functions fulfilling $\Delta u = 0$

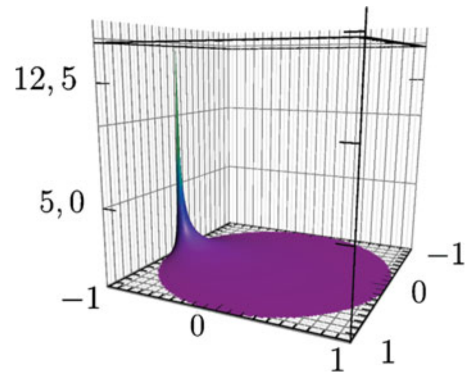


Fig. 1. The Poisson kernel in dependence on $\boldsymbol{x}$ for $d = 2$ and $\boldsymbol{y} = (-1, 0) \in \partial B^2$.

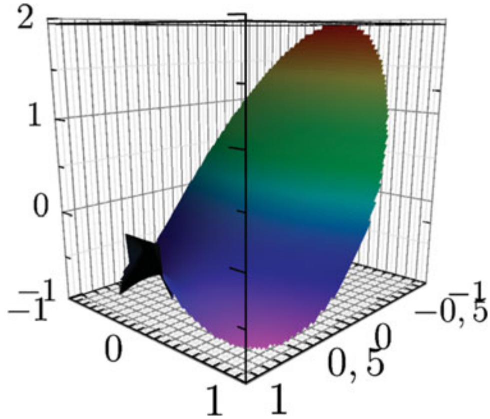


Fig. 2. The function $u(x, y) = \mathrm{Re}\left((1 - x - iy) \cdot \ln(1 - x - iy)\right)$ is a solution of a two-dimensional Dirichlet problem for Laplace's Equation to polynomial-time computable boundary data and illustrates that such a function need not be $C^1(\overline{B})$.

on an open set are also known as harmonic functions and well investigated. In particular, it is known that these functions are real analytic on their domain.

In view of Fact 2.7.2, we want to polynomially bound the modulus of continuity of u. According to Example 2.5, this follows immediately if $u \in C^1(\overline{B})$. However, the latter is in general not the case:

Example 3.2. For $d = 2$, it is easy to see that the real part of any holomorphic function satisfies Laplace's Equation. Therefore, shifting the function from Example 2.8b, extending it to a holomorphic function on the unit disc and then taking the real part yields a solution u to polynomial-time computable boundary data, whose derivative does not continuously extend to the boundary of the unit disc (cf. Figure 2).

Instead, we will employ the following well-known gradient estimate towards the boundary:

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

Fact 3.3. For any solution u of Equation (L), we have

$$\|Du(\boldsymbol{x})\| \leqslant \frac{d \cdot \|g\|_\infty}{\operatorname{dist}(\boldsymbol{x}, \partial B)}.$$

Here Du denotes the gradient, that is the vector containing the partial derivatives $D_i u$ and $\operatorname{dist}(\boldsymbol{x}, \partial B) := \inf\{\|\boldsymbol{x}-\boldsymbol{y}\| \mid \boldsymbol{y} \in \partial B\}$ is the usual distance function. A proof can be found in Gilbarg and Trudinger (2001, Theorem 2.10). We now conclude that u allows for a polynomial modulus of continuity whenever g does:

Lemma 3.4. Let C be an integer such that $C \geqslant \operatorname{lb}\left(\frac{2\|g\|_\infty^2}{\lambda(B)}\right) + d + 2$. If μ is a modulus of continuity of $g \in C(\partial B)$, then the function

$$\nu(n) := d\mu(n+2) + 2n + C$$

is a modulus of continuity of the unique solution u of Equation (L).

Our proof quantitatively refines the standard one underlying Fact 3.1.

Proof. We verify that ν is a modulus of continuity of u. So take arbitrary $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ such that $\|\boldsymbol{x}-\boldsymbol{y}\| \leqslant 2^{-\nu(n)}$. Distinguish the cases that both points are close to the boundary and that both points lie well within B. For this set

$$\begin{aligned} t(n) &:= d\mu(n+2) + n + \operatorname{lb}\left(\frac{2\|g\|_\infty}{d \cdot \lambda(B)}\right) + d + 1 \\ &\leqslant \nu(n) - \operatorname{lb}(d\|g\|_\infty) - n - 1 \qquad (8) \\ &\leqslant \nu(n) - 1. \qquad (9) \end{aligned}$$

Note that Equation (9) and $\|\boldsymbol{x}-\boldsymbol{y}\| \leqslant 2^{-\nu(n)}$ together imply that either $\|\boldsymbol{x}\|, \|\boldsymbol{y}\| \leqslant 1-2^{-t(n)}$ or $\|\boldsymbol{x}\|, \|\boldsymbol{y}\| \geqslant 1-2^{-t(n)+1}$ or both.

Case $\|\boldsymbol{x}\|, \|\boldsymbol{y}\| \leqslant 1-2^{-t(n)}$: In this case, we can apply Fact 3.3 and use Equation (8) to obtain

$$\begin{aligned} |u(\boldsymbol{x}) - u(\boldsymbol{y})| &\leqslant \sup\left\{\|(Du)(\boldsymbol{x})\| \mid \boldsymbol{x} \in B_{1-2^{-t(n)}}(0)\right\} \cdot \|\boldsymbol{x}-\boldsymbol{y}\| \\ &\leqslant \frac{d \cdot \|g\|_\infty}{2^{-t(n)}} \|\boldsymbol{x}-\boldsymbol{y}\| \;\leqslant\; d \cdot \|g\|_\infty 2^{-\nu(n)+t(n)} \;\leqslant\; 2^{-n}. \end{aligned}$$

Case $\|\boldsymbol{x}\|, \|\boldsymbol{y}\| \geqslant 1-2^{-t(n)+1}$: Consider the element $\boldsymbol{z} := \frac{\boldsymbol{x}+\boldsymbol{y}}{\|\boldsymbol{x}+\boldsymbol{y}\|} \in \partial B$ and observe that

$$\begin{aligned} \|\boldsymbol{x}-\boldsymbol{z}\| &= \left\|\boldsymbol{x} - \frac{\boldsymbol{x}+\boldsymbol{y}}{\|\boldsymbol{x}+\boldsymbol{y}\|}\right\| = \left\|\frac{\boldsymbol{x}-\boldsymbol{y}}{2} + \frac{\boldsymbol{x}+\boldsymbol{y}}{2} - \frac{\boldsymbol{x}+\boldsymbol{y}}{\|\boldsymbol{x}+\boldsymbol{y}\|}\right\| \\ &\leqslant 2^{-\nu(n)-1} + \frac{2-\|\boldsymbol{x}+\boldsymbol{y}\|}{2} \leqslant 2^{-\nu(n)-1} + 2^{-t(n)+1} + 2^{-\nu(n)-1} \\ &\leqslant 2^{-t(n)+2} \leqslant 2^{-\mu(n+2)-1}, \end{aligned}$$

and therefore for any $\boldsymbol{t} \in \partial B$ with $\|\boldsymbol{t}-\boldsymbol{z}\| \geqslant 2^{-\mu(n+2)}$

$$\|\boldsymbol{x}-\boldsymbol{t}\| = \|\boldsymbol{x}-\boldsymbol{z}-(\boldsymbol{t}-\boldsymbol{z})\| \geqslant \big|\|\boldsymbol{t}-\boldsymbol{z}\| - \|\boldsymbol{x}-\boldsymbol{z}\|\big| \geqslant 2^{-\mu(n+2)-1}. \qquad (10)$$

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

Since the unique solution of Equation (L) with constant boundary condition $u|_{\partial B} \equiv 1$ is given by the constant function $u \equiv 1$, it must hold

$$\int_{\partial B} K(\boldsymbol{x},\boldsymbol{t})\, d\sigma(\boldsymbol{t}) = 1 \tag{11}$$

for any $\boldsymbol{x} \in B$. Therefore, and since $u|_{\partial B} = g$ and thus $u(\boldsymbol{z}) = g(\boldsymbol{z})$, we have

$$\begin{aligned} |u(\boldsymbol{x}) - u(\boldsymbol{z})| &= \left| \int_{\partial B} K(\boldsymbol{x},\boldsymbol{t}) \cdot \big(g(\boldsymbol{t}) - g(\boldsymbol{z})\big)\, d\sigma(\boldsymbol{t}) \right| \\ &\leqslant \int_{\partial B} K(\boldsymbol{x},\boldsymbol{t}) |g(\boldsymbol{t}) - g(\boldsymbol{z})|\, d\sigma(\boldsymbol{t}) \\ &= \int_{\partial B \cap B_{2^{-\mu(n+2)}}(\boldsymbol{z})} K(\boldsymbol{x},\boldsymbol{t}) |g(\boldsymbol{t}) - g(\boldsymbol{z})|\, d\sigma(\boldsymbol{t}) \\ &\quad + \int_{\partial B \cap B_{2^{-\mu(n+2)}}(\boldsymbol{z})^c} K(\boldsymbol{x},\boldsymbol{t}) |g(\boldsymbol{t}) - g(\boldsymbol{z})|\, d\sigma(\boldsymbol{t}). \end{aligned}$$

The first of these integrals can be estimated by using that μ is a modulus of continuity for g and Equation (11). Plugging the definition of the Poisson Kernel K into the second term, using Equation (10) and estimating the remaining integral leads to

$$\begin{aligned} |u(\boldsymbol{x}) - u(\boldsymbol{z})| &\leqslant 2^{-n-2} + \frac{2\|g\|_\infty}{d \cdot \lambda(B)} \frac{1 - \|\boldsymbol{x}\|^2}{2^{-d\mu(n+2)-d}} \\ &\leqslant 2^{-n-2} + \frac{2\|g\|_\infty}{d \cdot \lambda(B)} 2^{-t(n)+1+d\mu(n+2)+d} \leqslant 2^{-n-1}. \end{aligned}$$

Since the same reasoning works with $\boldsymbol{x}$ replaced by $\boldsymbol{y}$, one finally gets

$$|u(\boldsymbol{x}) - u(\boldsymbol{y})| \leqslant |u(\boldsymbol{x}) - u(\boldsymbol{z})| + |u(\boldsymbol{y}) - u(\boldsymbol{z})| \leqslant 2^{-n}.$$

□

Lemma 3.4 is effective in the sense that it allows to construct a polynomial-time oracle machine computing a modulus of continuity of the solution u when given oracle access to a modulus of continuity of g (and an upper bound to the supremum norm of g). We will expand on such aspects of uniform computation in the conclusion in Section 7.

3.2. *Poisson's equation*

We now turn to the homogeneous Dirichlet problem for Poisson's equation on the unit ball:

$$\Delta u = f \text{ in } B, \quad u|_{\partial B} = 0. \tag{P}$$

Note that for this equation to make sense, f needs to be defined only on the open unit ball. However, in case f is polynomial-time computable, Fact 2.7 implies that f extends continuously to the closed unit ball. The following considerations will thus frequently suppose $f \in C\left(\overline{B}\right)$.

Again, we want to investigate the solution operator $f \mapsto u$. As mentioned before, Petrini's example shows that $f \in C\left(\overline{B}\right)$ may lead to so-called weak solutions $u \notin C^2(B)$.

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

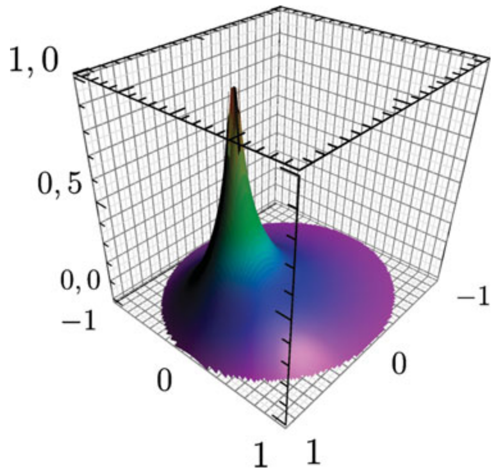


Fig. 3. The Green's function for $d = 2$ in dependence of $\boldsymbol{x}$ and with $\boldsymbol{y} = \left(\frac{1}{2}, 0\right) \in B^2$.

However we shall show that, for f admitting a polynomial modulus of continuity, the weak solution will actually be a (classical) solution; and the reader not familiar with this concept may simply understand 'weak solution corresponding to f' to mean the function w according to the following solution formula (compare for instance (Gilbarg and Trudinger 2001)):

$$w(\boldsymbol{x}) = \int_B G(\boldsymbol{y}, \boldsymbol{x}) \cdot f(\boldsymbol{y})\, d\boldsymbol{y} \tag{W}$$

with

$$G(\boldsymbol{y}, \boldsymbol{x}) = \tilde{\Gamma}(\|\boldsymbol{x} - \boldsymbol{y}\|) - \tilde{\Gamma}\left(\|\boldsymbol{x}\| \left\|\boldsymbol{y} - \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|^2}\right\|\right), \tag{G}$$

and

$$\tilde{\Gamma}(r) = \begin{cases} -\frac{1}{2\pi}\ln(r) & \text{if } d = 2, \\ \frac{1}{d\cdot(d-2)\cdot\lambda(B)\cdot r^{d-2}} & \text{if } d > 2 \end{cases}$$

(cf. Gilbarg and Trudinger (2001, Sections 2.4 and 2.5)). In Figure 3, the function G is depicted for the case $d = 2$, and in dependence of $\boldsymbol{x}$ for a fixed $\boldsymbol{y}$.

These integrals, taken from potential theory, make sense for any integrable and bounded function f. A reader familiar with potential theory may recognize the radially symmetric extension $\Gamma(\boldsymbol{x}) := \tilde{\Gamma}(\|\boldsymbol{x}\|)$ of $\tilde{\Gamma}$ as the *fundamental solution* of the d-dimensional Laplacian, and G as the *Green's function* obtainable from the fundamental solution by the method of image charges. Again, the reader not familiar with the underlying concepts may for our purpose simply consider them as names. We shall exploit only that (W) provides a concrete solution formula. Note however that, while $w \in C^1(B)$ (cf. the proof of Theorem 3.8), its second derivatives need not exist.

Fact 3.5. Whenever f is bounded and integrable and the corresponding weak solution is twice continuously differentiable, it is a solution of the homogeneous Dirichlet problem for Poisson's Equation (P).

A proof can for example be found in Gilbarg and Trudinger (2001).

It will often make sense to divide the weak solution w into the two summands appearing in Equation (G):

$$w(\boldsymbol{x}) = \underbrace{\int_B \Gamma(\boldsymbol{x}-\boldsymbol{y}) \cdot f(\boldsymbol{y})\, d\boldsymbol{y}}_{=:v(\boldsymbol{x})} - \underbrace{\int_B \tilde{\Gamma}\left(\|\boldsymbol{x}\| \cdot \left\|\boldsymbol{y} - \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|^2}\right\|\right) \cdot f(\boldsymbol{y})\, d\boldsymbol{y}}_{=:w_0(\boldsymbol{x})},$$

where v is also called the *Newtonian potential*. One can show that whenever f is integrable the Newtonian potential will be continuously differentiable on the whole space (Gilbarg and Trudinger 2001, Lemma 4.1). In particular, $v \in C^1\left(\overline{B}\right)$ and both v and its restriction to ∂B have a linear modulus of continuity. It is not hard to see that w_0 is actually the solution of Laplace's problem with boundary condition $v|_{\partial B}$, and Lemma 3.4 implies that w_0 also has a linear modulus of continuity. Thus w has a linear modulus of continuity, too. We will need this to show polynomial-time computability of the solution and list it as a lemma for reference.

Lemma 3.6. Whenever f is integrable and bounded, the corresponding weak solution has a linear modulus of continuity.

In Section 6, it will be convenient to know more about the weak solutions corresponding to a more restricted class of functions. There are well-known results of this kind. For example, it is known that w will be twice continuously differentiable whenever f is Hölder continuous. As we have seen in Example 2.5, this corresponds to functions with linear modulus of continuity. Our next goal is to show more generally that w is still twice continuously differentiable if f has a polynomial modulus of continuity. For this, we will need the following simple lemma:

Lemma 3.7. The series

$$\sum_{m=1}^{\infty} P(m) \cdot 2^{-m}$$

converges absolutely for any polynomial P.

Proof. Using l'Hôspital's rule it is easy to see that the function $h : [0;\infty) \to \mathbb{R},\ x \mapsto P(x)2^{-\frac{x}{2}}$ is bounded. Thus, we have

$$\sum_{m=n}^{\infty} |P(m) \cdot 2^{-m}| = \sum_{m=n}^{\infty} |P(m) \cdot 2^{-\frac{m}{2}}| \cdot 2^{-\frac{m}{2}} \leqslant \|h\|_\infty \cdot \sum_{m=0}^{\infty} (1/\sqrt{2})^m < \infty.$$

□

Lemma 3.8. If f has a polynomial modulus of continuity, the weak solution w is twice continuously differentiable in B.

Together with Fact 3.5, this shows that the weak solution is the wanted solution of the homogeneous Dirichlet Problem for Poisson's Equation in the case we are interested in.

Proof of the differentiability. As mentioned before, w_0 is the solution of a Dirichlet problem for Laplace's Equation and as such analytic on B.

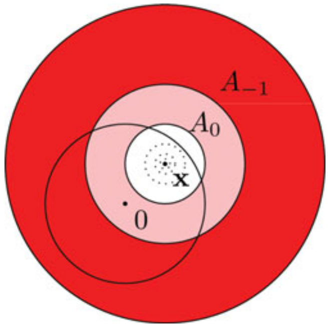


Fig. 4. The sets A_n for $d = 2$ and $\boldsymbol{x} = \left(\frac{1}{2}, \frac{1}{2}\right)$.

Thus, it suffices to show that the Newtonian potential

$$v : B \to \mathbb{R}, \quad \boldsymbol{x} \mapsto \int_B \Gamma(\boldsymbol{x} - \boldsymbol{y}) \cdot f(\boldsymbol{y})\, d\boldsymbol{y}$$

is twice continuously differentiable. It is well known that this function will be once continuously differentiable (Gilbarg and Trudinger 2001, Lemma 4.1).

By an observation by Morera which originates from Morera (1887), the Newtonian potential will be twice differentiable, whenever the integral

$$\int_B \frac{|f(\boldsymbol{x}) - f(\boldsymbol{y})|}{\|\boldsymbol{x} - \boldsymbol{y}\|^d}\, d\boldsymbol{y}$$

converges for any $\boldsymbol{x} \in B$. This can easily be seen to be true by re-evaluating the standard proof that the Newtonian potential of locally Hölder continuous functions is twice differentiable (see for example Gilbarg and Trudinger (2001, Lemma 4.2)).

To see that the integral is finite if there is a polynomial modulus of continuity, we divide the unit ball into spheres of finite thickness around $\boldsymbol{x}$

$$B = \bigcup_{n=-1}^{\infty} (B \cap \underbrace{\left(B_{2^{-\mu(n)}}(\boldsymbol{x}) \setminus B_{2^{-\mu(n+1)}}(\boldsymbol{x})\right)}_{=:A_n}),$$

where the convention $\mu(-1) = -1$ is made (cf. Figure 4), and estimate it by an infinite sum:

$$\begin{aligned}\int_B \frac{|f(\boldsymbol{x}) - f(\boldsymbol{y})|}{\|\boldsymbol{x} - \boldsymbol{y}\|^d}\, d\boldsymbol{y} &\leqslant \sum_{n=0}^{\infty} 2^{-n} \cdot \int_{A_n} \frac{1}{\|\boldsymbol{x} - \boldsymbol{y}\|^d}\, d\boldsymbol{y} \\ &= \ln(2)d \cdot \lambda(B) \cdot \sum\nolimits_{n=0}^{\infty} \left(\mu(n+1) - \mu(n)\right) \cdot 2^{-n}.\end{aligned}$$

This sum is finite by Lemma 3.7. □

4. The complexity of integration

Computational Complexity Theory is famous, among others, for its Millennium Prize $\mathcal{P}$ versus $\mathcal{NP}$ problem and several similar questions concerning inclusions of complexity classes generally believed to be strict but notoriously hard to prove so. Similarly to

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

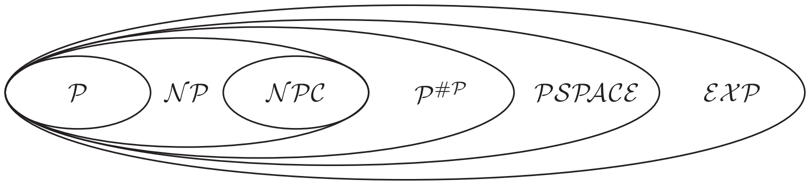


Fig. 5. The inclusion relation between the complexity classes.

Number Theory with the Riemann Hypothesis, results are therefore usually conditional: If $\mathcal{P} \neq \mathcal{NP}$, then the Boolean satisfiability problem cannot be solved within polynomial time – and vice versa. More precisely, SAT belongs to the class $\mathcal{NPC}$ of 'hardest' problems in $\mathcal{NP}$. Put differently, although no-one really knows the complexity of SAT, we do know that it coincides with the complexity of many, many other natural decision problems (Garey and Johnson 1979).

In particular, the famous result by Friedman and Ko, classifying the complexity of integration, is of this form. Recall that $\#\mathcal{P}$ can be phrased as the class of functions $\varphi : \{0,1\}^* \to \mathbb{N}$ such that there exists a polynomial-time decidable set $V \subseteq \{0,1\}^*$, called the *verifier set*, and a polynomial P such that

$$\varphi(a) \;=\; \#\left\{ b \in \{0,1\}^{P(l(a))} \,\middle|\, \langle a, b\rangle \in V \right\}. \tag{14}$$

So, each $\mathcal{NP}$ problem asking for the existence of a witness verifiable within polynomial time, leads to a $\#\mathcal{P}$ function computing the number of witnesses. For the $\mathcal{NPC}$ problem SAT, for example, this function computes the number of satisfying assignments of a boolean formula. Following this line of thought it is easy to see that $\mathcal{FP} \subseteq \#\mathcal{P}$ and that $\mathcal{FP} = \#\mathcal{P}$ implies $\mathcal{P} = \mathcal{NP}$. Moreover, Toda's Theorem (Toda 1991) implies that the entire polynomial-time hierarchy lies beneath $\mathcal{P}^{\#\mathcal{P}}$: the class of decision problems solvable in polynomial time using some $\varphi \in \#\mathcal{P}$ as oracle. Together with $\mathcal{EXP}$ and $\mathcal{PSPACE}$, the classes of decision problems solvable in exponential time resp. polynomial space, one arrives at the usual complexity picture (cf. Figure 5).

We will now state and prove a variant of Fact 1.1 from the Introduction. This involves a notion of polynomial-time computability for sequences of real functions.

Definition 4.1. A sequence $(f_m)_{m\in\mathbb{N}}$ of functions $f_m : \Omega \subseteq \mathbb{R}^d \to \mathbb{R}$ with common domain $\Omega \in \left\{ \overline{B}^d, [0;1]^d \right\}$ is called *polynomial-time computable* if both of the following conditions are fulfilled:

— There is a polynomial-time computable function $\phi : \{0,1\}^* \to \mathbb{D}$ such that whenever $\boldsymbol{q}$ is a dyadic point of Ω and $n, m \in \mathbb{N}$ we have

$$|f_m(\boldsymbol{q}) - \phi(\langle 1^m, [\![\boldsymbol{q}]\!], 1^n\rangle)| \leqslant 2^{-n}.$$

— The sequence $(f_m)_{m\in\mathbb{N}}$ admits a polynomial modulus of continuity, that is, a polynomial μ such that for every m the mapping $n \mapsto \mu(n+m)$ constitutes a modulus of continuity of f_m.

The latter condition relaxes equicontinuity by permitting a polynomial dependence of the modulus of continuity on the function index. Note also that the algorithm computing

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

ϕ is granted time polynomial in the value, that is, the unary length, of the index m (cf. Labhalla et al. (2001, Definition 2.2.9)).

The following finding will be one of the key tools in order to establish the main result of this paper. It is listed as a scholium as it follows by slightly adjusting the standard proof of Fact 1.1. A fully uniform generalization appeared in Kawamura (2011, Section 4.3.2), building on second-order complexity theory; see Section 7.

Scholium 4.2. For any fixed natural numbers $d, d' \geqslant 1$ the following are equivalent:

i. $\mathcal{FP} = \#\mathcal{P}$.

ii. For every polynomial-time computable sequence $(f_m)_{m\in\mathbb{N}}$ of functions $f_m : [0;1]^d \times [0;1]^{d'} \to \mathbb{R}$, the following sequence $(g_m)_{m\in\mathbb{N}}$ again is polynomial-time computable:

$$g_m : [0;1]^d \to \mathbb{R}, \quad \boldsymbol{x} \mapsto \int_{[0;1]^{d'}} f_m(\boldsymbol{x},\boldsymbol{y})\, d\boldsymbol{y}.$$

iii. For every polynomial-time computable smooth function $f : [0;1] \times [0;1] \to \mathbb{R}$, the parameter integral $[0;1] \ni x \mapsto \int f(x,y)\, dy$ is again polynomial-time computable.

We will mainly be concerned with the proof that from (i) follows (ii), as this is the statement that will be employed in the proofs in Section 5.

Proof of (i)$\Rightarrow$ (ii). First note that any modulus of continuity μ of $(f_m)_{m\in\mathbb{N}}$ is also one of $(g_m)_{m\in\mathbb{N}}$: If we assume $\|\boldsymbol{x}-\boldsymbol{x}'\| \leqslant 2^{-\mu(n+m)}$, then also $\left\|\binom{\boldsymbol{x}}{\boldsymbol{y}} - \binom{\boldsymbol{x}'}{\boldsymbol{y}}\right\| = \|\boldsymbol{x}-\boldsymbol{x}'\| \leqslant 2^{-\mu(n+m)}$ and therefore

$$|g_m(\boldsymbol{x}) - g_m(\boldsymbol{x}')| \;\leqslant\; \int_{[0;1]^{d'}} |f_m(\boldsymbol{x},\boldsymbol{y}) - f_m(\boldsymbol{x}',\boldsymbol{y})|\, d\boldsymbol{y} \;\leqslant\; \int_{[0;1]^{d'}} 2^{-n}\, d\boldsymbol{y} \;=\; 2^{-n}.$$

Let ϕ be the function computing the values of $(f_m)_{m\in\mathbb{N}}$ on dyadic arguments in polynomial time; that is, for a vector of dyadics $(\boldsymbol{q},\boldsymbol{s}) \in [0;1]^d \times [0;1]^{d'}$

$$|f_m(\boldsymbol{q},\boldsymbol{s}) - \phi(\langle 1^m, [\![\tbinom{\boldsymbol{q}}{\boldsymbol{s}}]\!], 1^n\rangle)| \leqslant 2^{-n}.$$

We want to approximate the sequence $(g_m)_{m\in\mathbb{N}}$. For this, we define a $\#\mathcal{P}$ function ψ that will take natural numbers n, m and a vector $\boldsymbol{q}$ of dyadics as input and then count how many squares of size related to n fit beneath the approximations of the function $\boldsymbol{x} \mapsto f_m(\boldsymbol{q},\boldsymbol{x})$. Towards this consider the polynomial-time decidable set

$$V := \left\{ \langle\langle [\![\boldsymbol{s}]\!], [\![t]\!]\rangle, \langle 1^m, [\![\boldsymbol{q}]\!], 1^n\rangle\rangle \;\middle|\; \begin{array}{l} \boldsymbol{s} \in \mathbb{D}^{d'}_{\mu(n)} \cap (0;1)^{d'},\ \boldsymbol{q} \in \mathbb{D}^{d}_{\mu(n)} \cap (0;1)^d, \\ t \in \mathbb{D}_n,\ 0 < t \leqslant \phi(\langle 1^m, [\![(\tbinom{\boldsymbol{q}}{\boldsymbol{s}})]\!], 1^n\rangle) \end{array} \right\}$$

(compare Figure 6). The corresponding $\#\mathcal{P}$ function

$$\psi(\langle 1^m, [\![\boldsymbol{q}]\!], 1^n\rangle) := \#\left\{ \langle [\![\boldsymbol{s}]\!], [\![t]\!]\rangle \in \{0,1\}^{P(n)} \;\middle|\; \langle\langle [\![\boldsymbol{s}]\!], [\![t]\!]\rangle, \langle 1^m, [\![\boldsymbol{q}]\!], 1^n\rangle\rangle \in V \right\},$$

where P is a polynomial such that $P(n) \geqslant |\langle\langle 1^{d\mu(n)}, 1^{d\mu(n)}\rangle, \langle 1^{\mu(n)}, 1^n\rangle\rangle|$, is as needed. Since we assume $\mathcal{FP} = \#\mathcal{P}$, this function will be computable in polynomial time. Now the function

$$\tilde{\phi}(\langle 1^m, [\![\boldsymbol{q}]\!], 1^n\rangle) := \frac{\psi(\langle 1^m, [\![\boldsymbol{q}]\!], 1^n\rangle)}{2^{n+\mu(n)}}$$

computes approximations the sequence $(g_m)_{m\in\mathbb{N}}$ in polynomial time. □

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

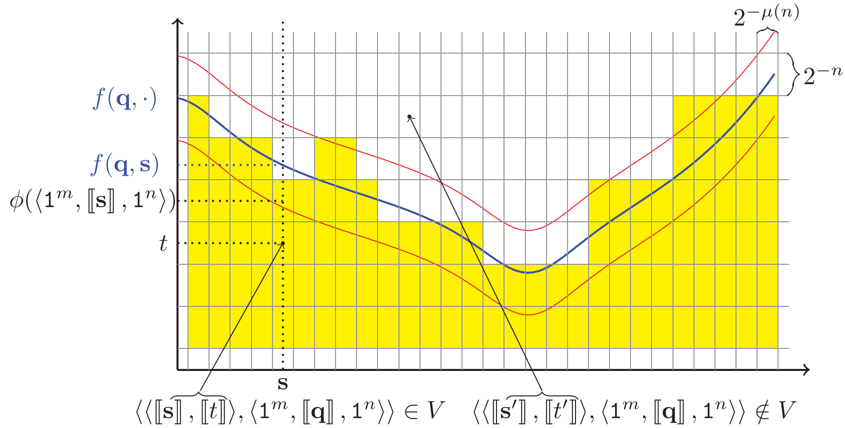


Fig. 6. Membership of the set V: The witnesses for some Element $\langle 1^m, [\![q]\!], 1^n\rangle$ are those pairs $\langle [\![s]\!], [\![t]\!]\rangle$ such that the corresponding square lies beneath the graph of the function section $f(q,\cdot)$ (that is up to finite approximation of the function values with precision n).

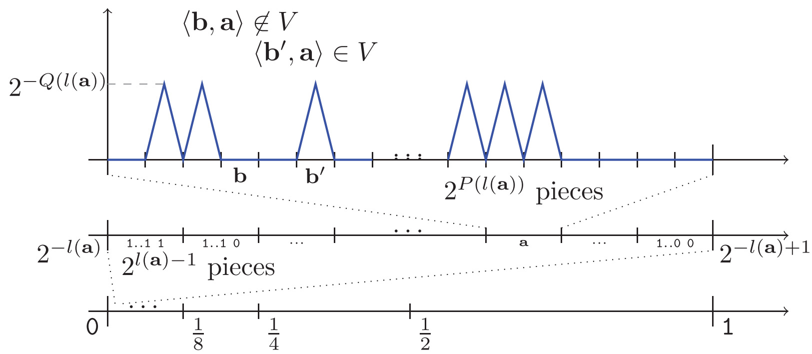


Fig. 7. The function $h_{P,V}$.

Section 5 will apply this to show that the solution operator of the Dirichlet problem for Poisson's equation carries polynomial-time computable functions to polynomial-time computable functions if $\mathcal{FP} = \#\mathcal{P}$. In Section 6, we will show that also the converse is true: If the solution operator preserves polynomial-time computability, then so does Riemann integration. The implication (ii)⇒(i) of Fact 1.1 fills the remaining gap of showing that this property of the integration operator implies $\mathcal{FP} = \#\mathcal{P}$. For sake of completeness, we give a very brief sketch of the proof for this given in Ko (1991):

Sketch of the proof of Fact 1.1 (ii)⇒(i). Consider some $\#\mathcal{P}$ function ψ. Then there is a polynomial P and a polynomial-time decidable set V, such that

$$\psi(\boldsymbol{a}) = \#\left\{ \boldsymbol{b} \in \{0,1\}^{P(|\boldsymbol{a}|)} \,\middle|\, \langle \boldsymbol{b}, \boldsymbol{a}\rangle \in V \right\}.$$

Now consider a 'bump function' $h_{P,V} : [0;1] \to \mathbb{R}$ as depicted in Figure 7. If we choose the function Q encoding the height of the bumps to be a polynomial and the degree high enough, this function will be Lipschitz continuous. Therefore, it has a linear modulus of

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

continuity. The functions values on dyadic numbers can effectively be computed, since it can be verified whether or not some b is a witness. Thus $h_{P,V}$ will be polynomial-time computable. Furthermore, $\psi(a)$ can be read from the binary expansion of the integral over the interval corresponding to a. This integral can be computed from the antiderivative. □

The ideas for the equivalence of Fact 1.1 (iii) are as follows: The implication (ii)⇒(iii) is obvious, and to establish the converse replace the triangle bump function in the above proof by smooth bump function. This is in particular worth mentioning as it constitutes a recurring theme in Real Complexity Theory: 'smoothness does not help' – only analyticity does (Müller 1987; Kawamura et al. 2014).

Also note that just like it was possible to adjust the proof of implication (i)⇒(ii) of Fact 1.1 to prove one direction of Scholium 4.2, the reverse direction can be modified to complete the proof: By compensating the absence of the ability to integrate over a specified interval with availability of at least one extra dimension, one can easily construct a function (that is a constant sequence) analogous to the one from the proof before.

5. Rigorous algorithmic solutions within $\#\mathcal{P}$

In this section, we will state and prove that the assumption $\mathcal{FP} = \#\mathcal{P}$, necessary for antiderivatives of polynomial-time computable functions to be polynomial-time computable, also allows to compute the solutions of the Dirichlet problem for Poisson's Equation in polynomial time whenever the data are polynomial-time computable. The notation in this section will be in accordance with Section 3.

As announced in the introduction, we treat two separate problems: Laplace's Equation and the homogeneous case for Poisson's Equation. The general approach will be very similar for both: showing polynomial-time computability using Fact 1.1. Half of that goal – polynomial bounds on the solutions' moduli of continuity – has already been achieved in Lemmata 3.4 and 3.6. To compute the solutions on dyadic points, we will use the explicit solution formulas from Section 3 to construct a polynomial-time computable sequence of approximations by truncating the unbounded integrands closer and closer to the singularity and then apply Scholium 4.2. The algorithms producing these sequences can then be used to compute the needed approximations by noting that the distance of a dyadic point of the unit ball from the boundary does not shrink too fast as the size of the encoding increases.

5.1. *Laplace's equation*

First, let us state the first third of the main result of this paper:

Theorem 5.1. If $\mathcal{FP} = \#\mathcal{P}$, then the unique solution of the general Dirichlet problem for Laplace's Equation (L) will be polynomial-time computable whenever g is.

Recall the solution formula from Section 3.1:

$$w : B \to \mathbb{R}, \quad \boldsymbol{x} \mapsto \int_{\partial B} K(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{y}) \, d\sigma(\boldsymbol{y}), \tag{PI}$$

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

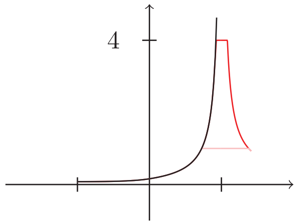


Fig. 8. Cross-section of Poisson kernel for $d = 2$: with singularity at $\boldsymbol{y} = (1, 1)/\sqrt{2} \in \partial B^2$ and truncated to a total continuous function.

where

$$K(\boldsymbol{x}, \boldsymbol{y}) = \frac{1 - \|\boldsymbol{x}\|^2}{d \cdot \lambda(B)\|\boldsymbol{x} - \boldsymbol{y}\|^d}. \tag{K}$$

If we want to apply Scholium 4.2, two difficulties arise: First, the integrand was always assumed to be polynomial-time computable; the Poisson Kernel K however, is not bounded and therefore not polynomial-time computable. Secondly, Scholium 4.2 refers to integrals of function sequences over squares while the domain we now want to integrate over the unit sphere.

As workaround we shall approximate the unbounded Poisson Kernel by truncating it close to the singularity and bound the integral error after transforming to spherical coordinates. Consider the sequence of functions

$$K_m : [-1;1]^d \times [-1;1]^d \to \mathbb{R}, \quad (\boldsymbol{x}, \boldsymbol{y}) \mapsto \begin{cases} \min\left\{ \frac{1-\|\boldsymbol{x}\|^2}{\|\boldsymbol{x}-\boldsymbol{y}\|^d}, \frac{1-\|\boldsymbol{x}\|^2}{2^{-dm}} \right\} & \text{if } \boldsymbol{x} \neq \boldsymbol{y} \\ \frac{1-\|\boldsymbol{x}\|^2}{2^{-dm}} & \text{if } \boldsymbol{x} = \boldsymbol{y} \end{cases}$$

(cf. Figure 8) and set

$$w_m(\boldsymbol{x}) := \int_{\partial B} K_m(\boldsymbol{x}, \boldsymbol{y}) \cdot g(\boldsymbol{y})\, d\sigma(\boldsymbol{y}).$$

Next, we list two lemmas which are the crucial ingredients for the proof of Theorem 5.1: The first one shows that w_m approximates w on a set big enough:

Lemma 5.2. Whenever $\boldsymbol{x} \in B_{1-2^{-m}}(0)$, we have

$$w_m(\boldsymbol{x}) = w(\boldsymbol{x}).$$

Proof. The requirement $\boldsymbol{x} \in B_{1-2^{-m}}(0)$ implies that for any $\boldsymbol{y} \in \partial B$

$$\|\boldsymbol{x} - \boldsymbol{y}\|^d \geqslant (1 - \|\boldsymbol{x}\|)^d \geqslant 2^{-dm}$$

and therefore from $\boldsymbol{x} \in B_{1-2^{-m}}(0)$ it follows that $K_m(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{x}, \boldsymbol{y})$ and for such $\boldsymbol{x}$ also

$$w_m(\boldsymbol{x}) = \int_{\partial B} K_m(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{y})\, d\sigma(\boldsymbol{y}) = \int_{\partial B} K(\boldsymbol{x}, \boldsymbol{y}) \cdot g(\boldsymbol{y})\, d\sigma(\boldsymbol{y}) = w(\boldsymbol{x}),$$

which is what we wanted. □

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

The second tool asserts that the sequence of approximations is polynomial-time computable if the assumption of Theorem 5.1 is met.

Lemma 5.3. If $\mathcal{FP} = \#\mathcal{P}$, then the sequence $(w_m)_{m\in\mathbb{N}}$ will be polynomial-time computable.

Proof. We aim to apply Scholium 4.2. For this, we transform into spherical coordinates, this will turn the integration over ∂B into an integration over a $d-1$ dimensional square. To show that the integrand sequence of the transformed integral is polynomial-time computable, we will first show that the sequence K_m is polynomial-time computable.

To see this, note that the function

$$h_m : [-1;1]^d \times [-1;1]^d \to \mathbb{R}, \quad (\boldsymbol{x}, \boldsymbol{y}) \mapsto \begin{cases} \min\left\{\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|^d}, \frac{1}{2^{-dm}}\right\} & \text{if } \boldsymbol{x} \neq \boldsymbol{y} \\ \frac{1}{2^{-dm}} & \text{if } \boldsymbol{x} = \boldsymbol{y} \end{cases}$$

can easily be seen to be Lipschitz continuous with Lipschitz constant $L_m = d \cdot 2^{(d+1)m}$. This means that $\mu(n) := (d+1)n + C$ is a modulus of continuity (compare the second item of Definition 4.1) for $(h_m)_{m\in\mathbb{N}}$ whenever $C > \mathrm{lb}(d)$. It is not hard to see that h_m can be efficiently approximated on dyadic inputs. Therefore, $(h_m)_{m\in\mathbb{N}}$ and due to $K_m = (1 - \|\boldsymbol{x}\|^2)h_m$ also $(K_m)_{m\in\mathbb{N}}$ forms a polynomial-time computable sequence.

Recall that the transformation to spherical coordinates in d dimensions is given by

$$\Phi(r, \theta_1, \ldots, \theta_{d-1})_i = \begin{cases} r \cdot \cos(\theta_i) \cdot \prod_{j=1}^{i-1} \sin(\theta_j) & \text{if } i < d-1 \\ r \cdot \sin(\theta_{d-1}) \prod_{j=1}^{i-2} \sin(\theta_j) & \text{if } i = d-1 \\ r \cdot \cos(\theta_{d-1}) \prod_{j=1}^{i-2} \sin(\theta_j) & \text{if } i = d \end{cases}$$

(cf. for instance Blumenson (1960)) with Jacobian determinant

$$|D\Phi|(r, \theta_1, \ldots, \theta_{d-1}) = r^{d-1} \cdot \prod_{j=1}^{d-2} \sin^{d-1-j}(\theta_j).$$

Note that all these functions are polynomial-time computable and that polynomial-time computability of the function

$$\tilde{g} : \theta := (\theta_1, \ldots, \theta_{d-1}) \mapsto g\big(\Phi(1, \theta_1, \ldots, \theta_{d-1})\big)$$

follows from closure under composition. Therefore, the function sequence

$$I_m : (\boldsymbol{x}, \theta) \mapsto K_m\big(\boldsymbol{x}, \Phi(1, \theta)\big) \cdot \tilde{g}(\theta) \cdot |D\Phi|(1, \theta)$$

is polynomial-time computable. Application of Scholium 4.2 to the transformed integral

$$w_m(\boldsymbol{x}) = \int_{\partial B} K_m(\boldsymbol{x}, \boldsymbol{y}) \cdot g(\boldsymbol{y})\, d\sigma(\boldsymbol{y}) = \int_{[0;2\pi]\times[0;\pi]^{d-2}} I_m(\boldsymbol{x}, \theta)\, d\theta$$

finally shows that the sequence $(w_m)_{m\in\mathbb{N}}$ is polynomial-time computable. □

Putting the above together, we can give a proof of the main result of this section.

Proof of Theorem 5.1. Let us assume that $\mathcal{FP} = \#\mathcal{P}$ and that we are given a polynomial-time computable function $g : \partial B \to \mathbb{R}$. By Fact 3.1 the unique solution u of Equation (L) is given by the continuous extension of w from Equation (PI). By Fact 2.7, we have

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

to specify a polynomial modulus of continuity of u and a polynomial-time function ϕ approximating the function values.

It follows from Lemma 3.4 that u has a polynomial modulus of continuity if g has one.

It remains to show that the values of w on dyadic points from the interior of the unit ball can be efficiently approximated. This can be done by putting the Lemmata 5.2 and 5.3 together. Since we assume $\mathcal{FP} = \#\mathcal{P}$, Lemma 5.3 guarantees that there is a polynomial computable function ψ computing approximations to the sequence $(w_m)_{m\in\mathbb{N}}$ on dyadic points.

Let $\boldsymbol{q} \in B$ be a vector of dyadics and let k be the least integer such that all of the components of $\boldsymbol{q}$ have a representation from $\mathbb{D}_k$. Then $\|\boldsymbol{q}\|^2 \in \mathbb{D}_{2k} \cap [0;1)$ and therefore

$$\|\boldsymbol{q}\| = \sqrt{1 + \|\boldsymbol{q}\|^2 - 1} \leqslant 1 + \frac{\|\boldsymbol{q}\|^2 - 1}{2} \in \mathbb{D}_{2k+1} \cap [0;1).$$

This implies that $\boldsymbol{q} \in B_{1-2^{-2(k+1)}}(0)$. According to Lemma 5.2 this means that u and w_{2k+2} coincide on $\boldsymbol{q}$. Furthermore, the function $k([\![\boldsymbol{q}]\!]) := 1^{2k+2}$ can be computed in polynomial time.

Thus the function

$$\phi(\langle [\![\boldsymbol{q}]\!], 1^n \rangle) := \psi(\langle 1^{2k+2}, [\![\boldsymbol{q}]\!], 1^n \rangle)$$

will compute the values of u on dyadic arguments in polynomial time. □

5.2. *Poisson's equation*

We state the second third of our main result:

Theorem 5.4. If $\mathcal{FP} = \#\mathcal{P}$, then the unique solution u of the homogeneous Dirichlet problem for Poisson's Equation (P) will be polynomial-time computable whenever f is.

The procedure to show this will be exactly the same as in the preceding subsection. Thus, recall the explicit solution formula from Section 3.2:

$$w(\boldsymbol{x}) = \int_B G(\boldsymbol{y}, \boldsymbol{x}) \cdot f(\boldsymbol{y})\, d\boldsymbol{y} \tag{W}$$

where G is defined in Equation (G).

We have already obtained a modulus of continuity for the solution w in Lemma 3.6. Upon trying to compute the solution on dyadic points, we meet the exact same complications we already encountered in the previous section: The integrand in (W) is unbounded (indeed has a singularity at $\boldsymbol{y} = \boldsymbol{x}$) and therefore not polynomial-time computable; furthermore integration is carried out over a ball instead of a cube. These difficulties will be dealt with in the exact same manner as before, but will cause more technical complications.

To avoid the singularities of the Green's function, we truncate the fundamental solution. Set

$$\tilde{\Gamma}_m : [0; d+2] \to \mathbb{R}, r \mapsto \begin{cases} \min\{\tilde{\Gamma}(2^{-m}), \tilde{\Gamma}(r)\} & \text{if } r \neq 0, \\ \tilde{\Gamma}(2^{-m}) & \text{if } r = 0. \end{cases}$$

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

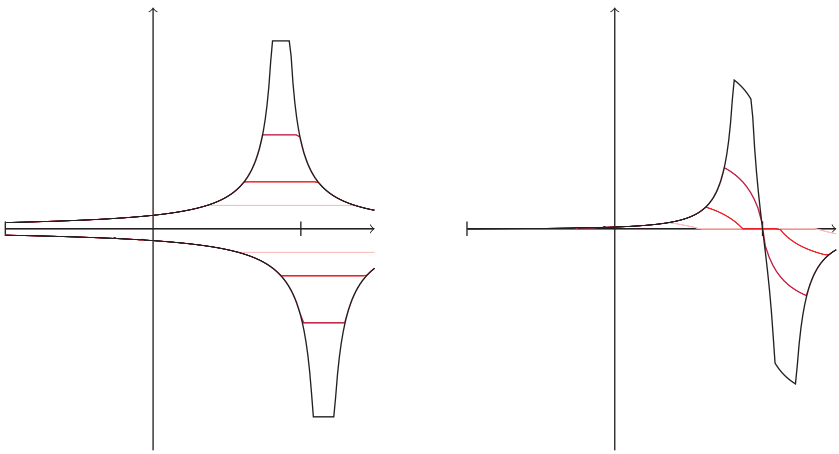


Fig. 9. Left: Cross-section of the Green's function G for $d = 3$ with singularity at $y = (1, 1, 1)/2$ and truncated at three levels; Right: We actually truncate the fundamental solution Γ whose difference then yields G.

We will need:

Lemma 5.5. The function sequence $\left(\tilde{\Gamma}_m\right)_{m\in\mathbb{N}}$ is computable in polynomial time.

Proof. It is easy to see that $\tilde{\Gamma}_m$ are Lipschitz continuous with Lipschitz constants

$$L_m = 2^{(d-1)m - \mathrm{lb}(d\cdot\lambda(B))}.$$

Since the dimension d is fixed, we can choose an integer upper bound C of $-\mathrm{lb}\big(d \cdot \lambda(B)\big)$ and the function $\mu(n) := (d-1)n + C$ will be a modulus of continuity for $(\tilde{\Gamma}_m)_{m\in\mathbb{N}}$.

It is straight forward to give algorithms computing the functions on dyadic arguments (for the case $d = 2$ compare Example 2.8b). □

In the spirit of Equation (G), we proceed to define a sequence $G_m : [-1;1]\times[-1;1] \to \mathbb{R}$ by

$$G_m(x, y) := \begin{cases} \tilde{\Gamma}_m(\|x-y\|) - \tilde{\Gamma}_m\left(\|y\| \left\| x - \frac{y}{\|y\|^2}\right\|\right) & \text{if } y \neq 0 \\ \tilde{\Gamma}_m(\|x\|) - \tilde{\Gamma}_m(1) & \text{if } y = 0 \end{cases}$$

(cf. Figure 9). Note that $\|x - y\| \leqslant \|x\| + \|y\| \leqslant 2\sqrt{d} \leqslant d + 2$ and $\|\|y\|(x - \frac{y}{\|y\|^2})\| \leqslant \|y\|\|x\| + 1 \leqslant d + 2$ and therefore G_m is well defined from the function sequence $(\tilde{\Gamma}_m)_{m\in\mathbb{N}}$ above.

Lemma 5.6. The sequence $(G_m)_{m\in\mathbb{N}}$ is polynomial-time computable.

Proof. It is easy to see that the functions

$$(x, y) \mapsto \|x - y\| \quad \text{and} \quad (x, y) \mapsto \begin{cases} \left\| \left(\|y\| x - \frac{y}{\|y\|} \right) \right\| & \text{if } y \neq 0 \\ 1 & \text{if } y = 0 \end{cases}$$

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

are polynomial-time computable. We briefly discuss how this can be seen for the second function: Since the function is Lipschitz continuous with Lipschitz constant 2, it has a linear modulus of continuity. Since all involved (component) functions are computable on $B \setminus B_{2^{-N}}(0)$ in time polynomial in the output precision and N we can proceed as follows: Given some argument $\boldsymbol{q}$ and a precision requirement n check whether $\|\boldsymbol{q}\|^2 \leqslant 2^{-2n+2}$. If it is, then compute approximations with the desired precision. If it is not, then return 1. Since we have the modulus of continuity, one can check that this leads to valid approximations in any case.

The modulus of continuity and the algorithm to compute the functions on dyadic arguments can now be easily obtained from those of these functions and $\tilde{\Gamma}_m$. □

Towards approximating the solution u consider the function sequence

$$w_m : B \to \mathbb{R}, \quad \boldsymbol{x} \mapsto \int_B G_m(\boldsymbol{y}, \boldsymbol{x}) \cdot f(\boldsymbol{y})\, d\boldsymbol{y}.$$

The following two lemmas are analogous to the Lemmas 5.2 and 5.3, and will be the key ingredients for the proof of Theorem 5.4.

Lemma 5.7. For $\boldsymbol{x} \in B_{1-2^{-m}}(0)$, we have

$$|w_m(\boldsymbol{x}) - w(\boldsymbol{x})| \leqslant \|f\|_\infty \cdot 2^{-m}.$$

Proof. Inserting the definitions of w_m and G_m leads to

$$\begin{aligned} |w_m(x) - w(x)| &= \left| \int_B (G_m(y,x) - G(y,x)) \cdot f(y)\, dy \right| \\ &\leqslant \left| \int_B (\tilde{\Gamma}_m(\|x-y\|) - \tilde{\Gamma}(\|x-y\|)) \cdot f(y)\, dy \right| \\ &+ \left| \int_B (\tilde{\Gamma}_m - \tilde{\Gamma}) \left(\|y\| \left\| x - \frac{y}{\|y\|^2} \right\| \right) \cdot f(y)\, dy \right|. \end{aligned}$$

Note that $\tilde{\Gamma}$ and $\tilde{\Gamma}_m$ do only differ on $(0; 2^{-m})$. We assumed $\|x\| \leqslant 1 - 2^{-m}$ and this implies

$$\|\boldsymbol{y}\| \cdot \left\| \boldsymbol{x} - \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|^2} \right\| \geqslant |\underbrace{\|\boldsymbol{y}\| \cdot \|\boldsymbol{x}\|}_{\leqslant 1} - 1| \geqslant 2^{-m},$$

thus the second integral is equal to zero. The first integral on the other hand can be estimated by transforming to spherical coordinates around $\boldsymbol{x}$:

$$|w_m(\boldsymbol{x}) - w(\boldsymbol{x})| \leqslant \|f\|_\infty \cdot d \cdot \lambda(B) \cdot \int_0^{2^{-m}} \tilde{\Gamma}(r) \cdot r^{d-1}\, dr.$$

The integration of $r^{d-1}\tilde{\Gamma}$ can be carried out explicitly by distinction of the cases $d = 2$ and $d > 2$ and leads to:

$$\int_0^{2^{-m}} r^{d-1} \cdot \tilde{\Gamma}(r)\, dr = \begin{cases} \frac{2^{-2m}(m+1)}{d \cdot \lambda(B)} & \text{if } d = 2 \\ \frac{2^{-2m-1}}{d \cdot (d-2) \cdot \lambda(B)} & \text{if } d > 2 \end{cases}.$$

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

Since $2^{-m}(m+1) \leqslant 1$ we have

$$\int_0^{2^{-m}} r^{d-1} \cdot \tilde{\Gamma}(r)\, dr \leqslant \frac{2^{-m}}{d \cdot \lambda(B)}$$

in both cases. Inserting this into the above inequality results in the desired inequality:

$$|w_m(\boldsymbol{x}) - w(\boldsymbol{x})| \leqslant \|f\|_\infty \cdot 2^{-m}.$$

□

Lemma 5.8. If $\mathcal{FP} = \#\mathcal{P}$, then the sequence $(w_m)_{m\in\mathbb{N}}$ will be polynomial-time computable.

Proof. The proof of this lemma is very similar to the proof of Lemma 5.3 and we keep it brief. From Lemma 5.6, we know that the sequence in the integrand is polynomial-time computable. We want to apply Scholium 4.2. To be able to do this we transform into spherical coordinates around the origin. To clarify that the integrand remains polynomial-time computable in this process, just note that the coordinate transformations and the functional determinant are polynomial-time computable. □

This enables us to prove the main result of this subsection:

Proof of Theorem 5.4. Assume that $\mathcal{FP} = \#\mathcal{P}$ and that f is polynomial-time computable. Since the Laplacian is linear and f is bounded, we can assume that $\|f\|_\infty \leqslant 1$.

By Theorem 3.8 the desired solution is given by the weak solution w from Equation (W) and twice continuously differentiable. We already argued that w has a polynomial modulus of continuity in Lemma 3.6. Note that w being twice continuously differentiable does not imply this, as it is only true on the interior of the ball.

To compute the values of w on dyadic arguments let $\boldsymbol{q} \in B$ be a vector of dyadics. Let k be the smallest number such that all of the components of $\boldsymbol{q}$ have a representation within $\mathbb{D}_k$. Analogously to the last proof from the preceding subsection, we find that $\|\boldsymbol{q}\| \leqslant 1 - 2^{-2k-1}$. And see that the function

$$k(\langle [\![\boldsymbol{q}]\!], 1^n\rangle) := 1^{\max\{2k+2, n+1\}}$$

is polynomial-time computable.

Apply Lemma 5.8 to find a polynomial-time computable function ψ approximating values of the sequence $(w_m)_{m\in\mathbb{N}}$. By Lemma 5.7 the function

$$\phi(\langle [\![\boldsymbol{q}]\!], 1^n\rangle) := \psi\big(\langle [\![\boldsymbol{q}]\!], 1^{n+1}, k(\langle [\![\boldsymbol{q}]\!], 1^n\rangle)\rangle\big)$$

approximates the function w on dyadic arguments in polynomial time. □

6. Optimality: $\#\mathcal{P}$-hardness of solving Poisson's Equation

The results from the previous section can be assembled to show that from Fact 1.1 (i), and therefore from any of the equivalent propositions, the statement of Theorem 1.2 (iv) follows. We now proceed to show the converse and third and final part of our main result: Assuming Theorem 1.2 (iv) we want to infer Fact 1.1 (ii), that is, that the antiderivative of a polynomial-time computable function on the unit interval is polynomial-time computable.

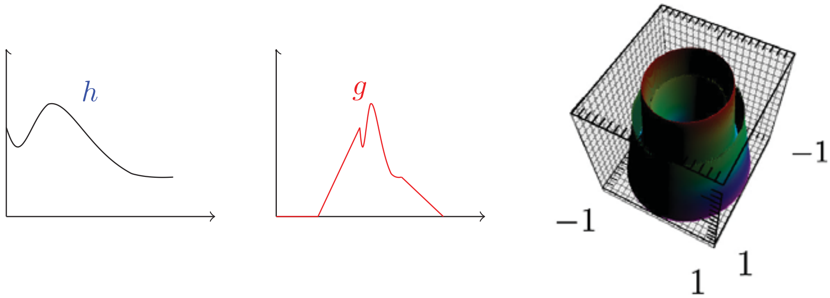


Fig. 10. The functions g and f for a sample function h in the case $d = 2$.

Proof of Theorem 1.2 (iv). Assume that the unique solution of Poisson's Equation is polynomial-time computable for any polynomial-time computable function f. We want to show that we can compute the antiderivative of an arbitrarily given polynomial-time computable function $h : [0;1] \to \mathbb{R}$ in polynomial time.

For this define a function $g : [0;1] \to \mathbb{R}$ by

$$g(x) := \begin{cases} 0 & \text{if } x \leqslant \frac{1}{4}, \\ 4h(0) \cdot (x - \frac{1}{4}) & \text{if } \frac{1}{4} < x \leqslant \frac{1}{2}, \\ h\big(4(x - \frac{1}{2})\big) & \text{if } \frac{1}{2} < x \leqslant \frac{3}{4}, \\ 4h(1) \cdot (1 - x) & \text{else} \end{cases}$$

and $f : B \to \mathbb{R}$ by

$$f(\boldsymbol{y}) := \begin{cases} \frac{g(\|\boldsymbol{y}\|)}{\|\boldsymbol{y}\|^{d-1}} & \text{if } \boldsymbol{y} \neq 0, \\ 0 & \text{if } \boldsymbol{y} = 0 \end{cases}$$

(cf. Figure 10). It is not hard to see that g and f are polynomial-time computable.

The function f is radially symmetric. Since the corresponding solution u is unique, it will also have to be radially symmetric. Rewriting Poisson's equation in spherical coordinates leads to

$$\frac{\partial \left(r^{d-1} \frac{\partial u}{\partial r}\right)}{\partial r}(\boldsymbol{y}) = g(\|\boldsymbol{y}\|).$$

Let $\tilde{u}$ be the function such that $u(\boldsymbol{y}) = \tilde{u}(\|\boldsymbol{y}\|)$. We will then have $\frac{\partial u}{\partial r}(\boldsymbol{y}) = \dot{\tilde{u}}(\|\boldsymbol{y}\|)$ and integrating both sides of the above equation from 0 to r results in

$$\int_0^r g(t)\,dt = r^{d-1} \cdot \dot{\tilde{u}}(r).$$

Taking the definition of g into consideration, we see that

$$\int_0^x h(t)\,dt = \int_0^{\frac{1}{2}+\frac{x}{4}} g(t)\,dt - \frac{h(0)}{2} = \left(\frac{1}{2} + \frac{x}{4}\right)^{d-1} \cdot \dot{\tilde{u}}\left(\frac{1}{2} + \frac{x}{4}\right) - \frac{h(0)}{2}.$$

Now since f is polynomial-time computable, u is twice continuously differentiable by Theorem 3.8 and polynomial-time computable by assumption. Thus, Proposition c shows that the partial derivatives of u restricted to $\overline{B}_{0.75}(0)$ and therefore also $\dot{\tilde{u}}$ restricted to

https://doi.org/10.1017/S096012951600013X Published online by Cambridge University Press

[0.5; 0.75] are polynomial-time computable. The above formula now shows that also the antiderivative of h is polynomial-time computable. □

Finally, we argue why we can even restrict to smooth functions f.

Proof of Theorem 1.2v. Note that the proof of Fact 1.1, (ii)⇒ (i) as sketched at the end of Section 4 not only proved the implication but explicitly constructed a polynomial-time computable function h such that its antiderivative being polynomial-time computable implies $\mathcal{FP} = \#\mathcal{P}$. If such a function is fed to the procedure above, we end up with a polynomial-time computable function f such that the solution u of the corresponding homogeneous Dirichlet problem for Poisson's equation being polynomial-time computable will imply $\mathcal{FP} = \#\mathcal{P}$. The proof of Fact 1.1, (iii)⇒ (i) was simply noting that the function h can also be chosen to be smooth. Furthermore, the support of this function can be chosen to be smaller than the whole interval. This will lead to a smooth function f. □

7. Conclusion and perspective

We have established matching upper and lower bounds on the computational complexity of solving the Dirichlet Problem for Poisson's Equation (1) on the Euclidean unit ball: For every polynomial-time computable right-hand side f and boundary condition g, the unique solution u is classical (i.e. C^2) and computable within $\#\mathcal{P}$; while, conversely, there exist polynomial-time f such that u is not polynomial-time computable unless $\mathcal{FP} = \#\mathcal{P}$. This contributes to a research program started by Harvey Friedman and Ker-I Ko of characterizing discrete complexity classes via numerical problems; and constitutes a first step towards a rigorous complexity theory of solving partial differential equations, thus refining recent results about their computability.

The upper bound follows from analyses of the Poisson Kernel and Green's Function and their singularities, permitting reductions to ordinary Riemann integration – and back, for the case of Poisson's Equation. In fact, we do not know whether solving the general Dirichlet problem for Laplace's Equation is also $\#\mathcal{P}$-hard. It is, however, $\#\mathcal{P}_1$-hard, starting from dimension two: Equation (PI) implies

$$u(0) = \frac{1}{d \cdot \lambda(B)} \int_{\partial B} g(\mathbf{y})\, d\sigma(\mathbf{y}).$$

Now recall from the introduction that definite integration corresponds to $\#\mathcal{P}_1$.

We emphasize the non-uniformity of our upper complexity bounds: Under the hypothesis $\mathcal{FP} = \#\mathcal{P}$, to every choice of polynomial-time algorithms $\mathcal{A}$ and $\mathcal{A}'$ computing f and g respectively, there exists an algorithm $\mathcal{B}$ computing the solution u within time polynomial in n, the output precision. Numerical practitioners are of course interested in the dependence of $\mathcal{B}$ and it's running time on $\mathcal{A}, \mathcal{A}'$ and their running time. In order to formally treat such important questions of uniformity also for (solution) operators, one of us and his PhD advisor have recently devised a promising framework (Kawamura and Cook 2012).

Future work will also explore domains more general than Euclidean balls such as squares, polyhedra and compact convex sets polynomial-time computable in the sense of Rösnick (2013).

This work is supported in part by *JSPS Kakenhi* projects 23700009 and 24106002, by *DFG* IRTG 1529, by *7th EU IRSES* project 294962 and by *DFG* project Zi 1009/4-1.

References

Blumenson, L.E. (1960). Classroom notes: A derivation of n-Dimensional spherical coordinates. *The American Mathematical Monthly* **67**(1) 63–66.

Bournez, O., Graça, D.S., Pouly, A. and Zhong, N. (2013). Computability and computational complexity of the evolution of nonlinear dynamical systems. In: *Proc. 9th Conference on Computability in Europe (CiE 2013)*, Springer LNCS vol. 7921, doi 10.1007/978-3-642-39053-1_2.

Brattka, V. and Yoshikawa, A. (2006). Towards computability of elliptic boundary value problems in variational formulation. *Journal of Complexity*, **22** 858–880.

Braverman, M. (2005). On the complexity of real functions. In: *Proc. 6th Annual IEEE Symposium on Foundations of Computer Science*, doi 10.1109/SFCS.2005.58.

Friedman, H. (1984). The computational complexity of maximization and integration. *Advances in Mathematics* **53**(1) 80–98.

Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York.

Gilbarg, D. and Trudinger, N.S. (2001). *Elliptic Partial Differential Equations of Second Order*, Classics in Mathematics. Springer-Verlag, Berlin. Reprint of the 1998 edition.

Grzegorczyk, A. (1957). On the definitions of computable real continuous functions. *Fundamenta Mathematicae* **44** 61–71.

Kawamura, A. (2010). Lipschitz continuous ordinary differential equations are polynomial-space complete. *Computational Complexity* **19**(2) 305–332.

Kawamura, A. (2011). *Computational Complexity in Analysis and Geometry*. PhD thesis, University of Toronto.

Kawamura, A. and Cook, S. (2012). Complexity theory for operators in analysis. *ACM Transactions in Computation Theory* **4**(2) Article 5.

Kawamura, A., Ota, H., Rösnick, C. and Ziegler, M. (2014). Computational complexity of smooth differential equations. *Logical Methods in Computer Science* **10**(1) 6.

Ko, K.-I. (1982). The maximum value problem and NP real numbers. *Journal of Computer and System Sciences* **24**(1) 15–35.

Ko, K.-I. (1990). Inverting a one-to-one real function is inherently sequential. In: *Feasible Mathematics (Ithaca, NY, 1989)*, Progress in Computer Science and Applied Logic, volume 9, Birkhäuser Boston, Boston, MA 239–257.

Ko, K.-I. (1991). *Complexity Theory of Real Functions*. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA.

Ko, K.-I. (1992). On the computational complexity of integral equations. *Annals of Pure and Applied Logic* **53**(3) 201–228.

Ko, K.-I. (1998). Polynomial-time computability in analysis. In: *Handbook of Recursive Mathematics, Vol. 2*, Studies in Logic and the Foundations of Mathematics, volume 139, North-Holland, Amsterdam 1271–1317.

Ko, K.-I. and Friedman, H. (1982). Computational complexity of real functions. *Theoretical Computer Sciences* **20**(3), 323–352.

Ko, K.-I. and Yu, F. (2008). On the complexity of convex hulls of subsets of the two-dimensional plane. In: *Proceedings of the 4th International Conference on Computability and Complexity in Analysis* (CCA 2007). *Electron. Notes Theor. Comput. Sci.* 202 Elsevier Sci. B. V., Amsterdam 121–135.

Labhalla, S., Lombardi, H. and Moutai, E. (2001). Espaces métriques rationnellement présentés et complexité. *Theoretical Computer Science* **250** 265–332. le cas de l'espace des fonctions réelles uniformément, continues sur un intervalle compact.

Lacombe, D. (1958). Sur les possibilités d'extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles. In: *Le raisonnement en mathématiques et en sciences expérimentales*, Colloques Internationaux du Centre National de la Recherche Scientifique, LXX, Editions du Centre National de la Recherche Scientifique, Paris 67–75.

Morera, G. (1887). Sulle derivate seconde della funzione potenziale di spazio. *Rendiconti* **20** 302–310.

Müller, N.T. (1987). Uniform computational complexity of Taylor series. In: *Proceeding 14th International Colloquium on Automata, Languages, and Programming*. *Springer-Verlag Lecture Notes in Computer Science* **267** 435–444.

Müller, N.T. (1995). Constructive aspects of analytic functions. In: *Proc. Workshop on Computability and Complexity in Analysis*, volume 190 of *InformatikBerichte*, FernUniversität Hagen 105–114.

Parberry, I. and Schnitger, G. (1988). Parallel computation with threshold functions. *Journal of Computer and System Sciences* **36** 278–302.

Pour-El, M.B. and Richards, J.I. (1989). *Computability in Analysis and Physics*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin.

Rösnick, C. (2013). Closed sets and operators thereon: Representations, computability and complexity. In: *Proceedings of the 10th International Conference on Computability and Complexity in Analysis (CCA)*, 8-10 July 2013, Nancy, France.

Sun, S., Zhong, N. and Ziegler, M. (2015). Computability of navier-stokes' equation. In: *Proceeding of the 11th Conference on Computability in Europe*, *Lecture Notes in Computer Science* **9136** 334–342.

Toda, S. (1991). PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing* **20**(5) 865–877.

Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* **42**(2) 230–265.

Weihrauch, K. (2000). *Computable Analysis. An Introduction.* Springer, Berlin/Heidelberg.

Weihrauch, K. and Zhong, N. (2002). Is wave propagation computable or can wave computers beat the Turing machine?. *Proceedings of London Mathematical Society* **85**(2) 312–332.

Weihrauch, K. and Zhong, N. (2005). Computing the solution of the Korteweg–de Vries equation with arbitrary precision on turing. *Theoretical Computer Science* **332**(1–3) 337–366.

Weihrauch, K. and Zhong, N. (2006). An algorithm for computing fundamental solutions. *SIAM Journal on Computing* **35**(6) 1283–1294.

Weihrauch, K. and Zhong, N. (2007). Computable analysis of the abstract Cauchy problem in a Banach space and its applications I. *Mathematical Logic Quarterly* **53**(4-5) 511–531.

Wienholtz, E., Kalf, H. and Kriecherbauer, T. (2009). *Elliptische Differentialgleichungen zweiter Ordnung*. Springer, Dordrecht. Eine Einführung mit historischen Bemerkungen. [An introduction with historical remarks].

Zhong, N. (1998). Derivatives of computable functions. *Mathematical Logic Quarterly* **44** 304–316.