

Electronic Notes in Theoretical Computer Science 221 (2008) 127–139

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

Technical Report: Computation on the Extended Complex Plane and Conformal Mapping of Multiply-connected Domains

Valentin V. Andreev,[1](#_bookmark0)

Dale Daniel[2](#_bookmark0) and Timothy H. McNicholl[3](#_bookmark0)

*Department of Mathematics Lamar University*

*Beaumont, Texas 77710 USA*

**Abstract**

We introduce a system for computation on the extended complex plane based on the Type-Two Effec- tivity approach to computable analysis. Included are computations on meromorphic functions, open sets, and closed sets. Applications to M¨obius transformations, boundaries of multiply connected domains, and conformal mapping of multiply connected domains are considered.

*Keywords:* Computable analysis, conformal mapping, Type-Two Effectivity

# Introduction

The Riemann mapping theorem states that all simply connected domains with more than one boundary point are conformally equivalent. This theorem made possible the study of conformal mappings of such simply connected domains onto one and the same *canonical domain*, namely, the unit disk.

There has been considerable interest in obtaining constructive proofs of the Riemann mapping theorem. Paul Koebe’s [[9](#_bookmark25)] approach to the simplification of the first complete proof of the theorem given by Carath´eodory is the cornerstone of the subsequent constructive proofs given by Cheng [[2](#_bookmark17)], Bishop and Bridges [[1](#_bookmark18)], and of the computable proof given by Hertling [[7](#_bookmark23)].

1 Email: [valentin.andreev@lamar.edu](mailto:valentin.andreev@lamar.edu)

2 Email: [dale.daniel@lamar.edu](mailto:dale.daniel@lamar.edu)

3 Email: [mcnichollth@my.lamar.edu](mailto:mcnichollth@my.lamar.edu)

1571-0661© 2008 Elsevier B.V. Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

doi:10.1016/j.entcs.2008.12.012

The study of conformal mappings of multiply connected domains onto canonical domains immediately encounters two major obstacles. The first is the fact that continuous mappings preserve connectivity and thus the possible canonical domain must be multiply connected. The second and the most important one is that two multiply connected domains of the same order of connectivity need not be confor- mally equivalent. For example, there is no conformal mapping between the annuli *r < |z| <* 1 and *R < |z| <* 1 whenever *r /*=*R* (for a simple proof of this fact, see [[12](#_bookmark28), p. 333]).

Every doubly connected domain is conformally equivalent to an annulus *μ <*

*|z| <* 1, where *μ* is a unique real number, 0 *< μ <* 1, called the *modulus* of the doubly connected domain. It can be shown that the conformal type of an *n*- connected domain (*n >* 2) is determined by 3*n −* 6 real numbers also called the *moduli* of the domain (for a derivation see [[12](#_bookmark28), p. 354, Exercise 13]).

For connectivity *n >* 2 there are various canonical domain types: the parallel slit domain, the circular slit domain, the radial slit domain, the circle with concentric circular slits, the circular ring with concentric circular slits, and the circular domain. Two circular domains of the same order of connectivity are conformally equivalent if and only if there is a Mo¨bius transfomation mapping one of them onto the other [[14](#_bookmark29), p. 426, Theorem IX.36].

In a paper preceding his proof of the Riemann mapping theorem Koebe [[8](#_bookmark24)] gave a very short outline of his method for construction of a conformal mapping of a multiply connected domain onto a circular domain. A detailed convergence proof of Koebe’s method was given by Gaier [[3](#_bookmark19)] almost fifty years later, and a more elaborate version of Gaier’s proof can be found in Henrici [[6](#_bookmark22), Theorem 17.7a]. Our aim is to present a computable proof of Koebe’s theorem based on Gaier’s approach.

This leads us to consider computation on the extended complex plane (*i.e.* the

Riemann sphere) which is the mathematically most convenient setting for the study of conformal mapping. We develop a model of computation on the extended com- plex plane based on the Type-Two Effectivity (TTE) approach to computable anal- ysis [[15](#_bookmark30)]. That is, we produce admissible representations of the extended complex plane, its open and closed sets, and the meromorphic functions. We then show that the fundamental operations such as the field operations, computations of zeros

and poles, Mo¨bius transformations, *etc.* are computable. In the last two sections,

we take up the problem of determining the precise amount of information neces- sary to construct a conformal map between two non-degenerate multiply connected domains.

Unless otherwise mentioned, all definitions and notations for computable anal- ysis are as in [[15](#_bookmark30)].

Proofs of many claims are omitted, but will appear in a future work.

# Background from complex analysis

Let Cˆ = C *∪ {∞}*. We identify C with R2.

## Definition 2.1 (Domain terminology)

1. A *domain* is an open connected subset of Cˆ.
2. A domain is *n-connected* if its complement has exactly *n* connected compo- nents.
3. A domain is *degenerate* if a component of its complement has fewer than two points.
4. A domain is *circular* if it is the result of removing one or more disjoint closed disks from Cˆ.

Each circular domain whose complement has at least two connected components is associated with two constants, *μ* and *δ*. These are defined as follows. Let *Dr*(*z*) denote the open disk with center *z* and radius *r*.

**Definition 2.2** Let *C* be an *n*-connected circular domain with *n ≥* 2, and let

*Dr*1 (*z*1)*,..., Drn* (*zn*) be the components of Cˆ *— C*. Define *μC* to be the reciprocal

of

min*{r >* 1 : *E*1 *≤ j, k ≤ n* (*j /*=*k Λ Drrj* (*zj*) *∩ Drrk* (*zk*) */*=*∅*)*}.*

**Definition 2.3** Let *C* be an *n*-connected circular domain with *n ≥* 2, and let

*Dr*1 (*z*1)*,..., Drn* (*zn*) be the components of Cˆ *— C*. Denote the boundary of *Dr* (*zj*)

*j*

by Γ*j*. Let Γ*k* denote the circle obtained by reflecting Γ*k* into Γ*j*. We define *δC* to

*j*

be

min*{d*(Γ*j,* Γ*k*) : 1 *≤ j, k ≤ n Λ j /*=*k}.*

*j*

Let *Ðn*(Cˆ) be the set of all *n*-connected domains.

The Riemann Mapping Theorem states (among other things) that every proper,

open, simply connected subset of the plane is conformally equivalent to the unit disk D. Such a set is a non-degenerate 1-connected domain. There is an extension of this result to *n*-connected subsets of the extended plane.

**Theorem 2.4** *If D is a non-degenerate n-connected domain, and if z*0 *∈ D, then* *there is a circular domain C and a conformal mapping f from D onto C such that f* (*z*0) = *∞.*

Unfortunately, when *n >* 1, it is not the case that all non-degenerate *n*-connected domains are conformally equivalent to the *same* circular region. However, we can get a weaker uniqueness result if we impose a restriction on the form of the conformal map.

**Theorem 2.5** *If D is a non-degenerate n-connected domain that contains ∞ but not* 0*, then there is a unique circular region CD such that there is a conformal map fD of D onto CD such that fD*(*z*) = *z* + *O*(*z−*1)*. Furthermore, this map is unique.*

Theorem [2.5](#_bookmark2) implies Theorem [2.4](#_bookmark1). Theorem [2.5](#_bookmark2) is proven in [[6](#_bookmark22)]. In Section [10](#_bookmark12), we will investigate the computable content of Theorem [2.5](#_bookmark2).

We will use winding numbers to compute interiors and exteriors of continuously differentiable simple closed curves. These are defined as follows.

**Definition 2.6** When *γ* is a rectifiable simple closed curve in C and *z ∈* Cˆ *—*ran(*γ*), let

*η*(*γ, z*) = 1 ∫

2*πi*

*γ*

*dζ*

*.*

*ζ — z*

*η*(*γ, z*) is called the *winding number of γ around z*.

If *γ* is a simple closed curve, then let *Int*(*γ*) denote the interior of *γ*, and let

*Ext*(*γ*) denote the exterior of *γ*. A proof of the following can be found in [[4](#_bookmark20)].

**Proposition 2.7** *Let γ be a rectiﬁable, simple closed curve.*

1. *For all z /*r*∈*an(*γ*)*, η*(*γ, z*) *is an integer.*
2. *Int*(*γ*) = *{z ∈* Cˆ
3. *Ext*(*γ*) = *{z ∈* Cˆ

: *η*(*γ, z*) */*=*}*0*.*

: *η*(*γ, z*) = 0*}.*

For computability purposes, we will approximate continuously differentiable curves with rational polygonal curves. These are defined as follows.

**Definition 2.8** (i) A curve *γ* : [*a, b*] *→* C is *polygonal* if it is continuous and there exist *t*0*,..., tk* such that *a* = *t*0 *< . . . < tk* = *b* and *γ* is linear on each of [*t*0*, t*1]*,...,* [*tk—*1*, tk*].

(ii) If, in addition, the coo¨rdinates of *γ*(*tj*) are rational for each *j*, then we will call

*γ* a *rational polygonal curve*.

The following lemma says that it is possible to simultaneously approximate a continuously differentiable simple closed curve and its derivative by a rational polygonal curve with arbitrary precision.

**Lemma 2.9** *Suppose γ* : [*a, b*] *→* C *is a simple closed curve with continuous deriva- tive. Then, for every ϵ >* 0*, there is a rational, polygonal, simple closed curve γ*0 : [*a, b*] *→* C *such that*

max*{|γ*(*t*) *— γ*0(*t*)*|* : *t ∈* [*a, b*]*} <ϵ*

sup*{|γ'*(*t*) *— γ'* (*t*)*|* : *t ∈* dom(*γ'* )*} <ϵ*

0 0

Informally speaking, the following lemma says it is possible to surround the connected components of the complement of an unbounded, non-degenerate *n*- connected domain with rational polygonal curves that are well-separated.

**Lemma 2.10** *Suppose D is a non-degenerate n-connected domain and ∞∈ D. Let C*1*,..., Cn be the components of the complement of D. Then, there exist rational polygonal curves γ*1*,..., γn such that the following hold.*

1. ran(*γj*) *⊆ D.*
2. *If j /*=*k, then* ran(*γj*) *∩* ran(*γk*) = *∅.*
3. *Cj ⊆ Int*(*γj*) *—* *k/*=*j Int*(*γk*)*.*

Using Lemma [2.10](#_bookmark3), we can prove the following, which will be used later to recognize the connected components of the complement of an unbounded, non- degenerate *n*-connected domain. Let *I*2 be the notation in Definition 4.1.2 of [[15](#_bookmark30)].

**Lemma 2.11** *Suppose D is a non-degenerate n-connected domain and ∞ ∈ D. Then, there exist words w*1*,..., wn and rational, polygonal, simple closed curves γ*1*,..., γn such that the following.*

* 1. ran(*γj*) *⊆ D.*
  2. *If j /*=*k, then* ran(*γj*) *∩* ran(*γk*) = *∅.*
  3. *I*2(*wj*) *⊆ Int*(*γj*) *— ∪k/*=*jInt*(*γk*)*.*
  4. *I*2(*wj*) *∩ ∂D /*=*∅.*

*Furthermore, if w*1*,..., wn, γ*1*,..., γn are such that (*[*i*](#_bookmark4)*)- (*[*iv*](#_bookmark6)*) hold, then it is possible to label the connected components of* Cˆ *— D, C*1*,..., Cn, so that Cj ⊆ Int*(*γj*) *—*

*∪k/*=*jInt*(*γk*)*.*

# Computable analysis

* 1. *Spaces to be considered and their default naming systems*

Suppose *X* is a topological space. We let *O*(*X*) be the set of open subsets of *X*. We let *C*(*X*) be the set of closed subsets of *X*.

The following table indicates some of our default naming systems. Definitions can be found in [[15](#_bookmark30)].

|  |  |
| --- | --- |
| Space | Default naming system |
| N | *ν*N |
| Q | *ν*Q |
| Σ*∗* | *Id*Σ*∗* |
| R | *ρ* |
| C | *ρ*2 |
| *O*(C) | *θ*2  *<* |

We introduce an admissible naming system for Cˆ by first building a computable topology on Cˆ. [4](#_bookmark7)

## Definition 3.1 (Computable topology on the extended plane)

* + 1. Let

C

*σ*ˆ = *{I*2(*w*) *| w ∈ dom*(*I*2)*} ∪ {*Cˆ *— I*2(*w*) *| w ∈ dom*(*I*2) *Λ* 0 *∈ I*2(*w*)*}.*

4 This is not the only way to set up computation on Cˆ. There is also the paper [[13](#_bookmark31)] in which computation on these spaces is defined by embedding Cˆ into R3.

Thus, *σ*Cˆ is a countable basis for the standard topology on Cˆ.

* + 1. For all *w ∈* dom(*I*2), let *ν*Cˆ (*⟨*0*, w⟩*) = *I*2(*w*). If 0 *∈ I*2(*w*), let *ν*Cˆ (*⟨*1*, w⟩*) =

Cˆ *— I*2(*w*).

* + 1. Let *S*Cˆ = (Cˆ*, σ*Cˆ *, ν*Cˆ ).

It follows that *S*Cˆ is a computable topological space. Let *δ*Cˆ = *δS*Cˆ . *δ*Cˆ

is our

default naming system for Cˆ. Informally speaking, a *δ*Cˆ -name of a *z ∈* Cˆ is an

exhaustive list of all basic neighborhoods to which *z* belongs. If *z* = *∞*, then this list will only contain neighborhoods of the form C *— I*2(*w*). But, if *z /*=*∞*, then this list will contains neighborhoods of *∞* as well as finite neighborhoods. These observations lead to the following.

**Proposition 3.2** *There is no computable F* :*⊆* Σ*ω → {*0*,* 1*} such that for all p ∈*

dom(*δ*Cˆ )*,*

We now turn to *O*(Cˆ).

*F* (*p*) = ⎧⎨ 1 *if δ*Cˆ (*p*) = *∞*

⎩ 0 *otherwise*

## Definition 3.3 (Computable topology on open subsets of the extended plane)

1. For all *w ∈* dom(*ν*Cˆ ), let

*νO*(Cˆ)(*w*)= *{U ∈ O*(Cˆ) *| ν*Cˆ (*w*) *⊆ U}.*

Then, let *σO*(Cˆ) = ran(*νO*(Cˆ)).

1. Let *SO*(Cˆ) = (*O*(Cˆ)*, σO*(Cˆ)*, νO*(Cˆ)).

Clearly, *SO*(Cˆ) is a computable topological space. Let *δO*(Cˆ) = *δSO*(Cˆ) . *δO*(Cˆ) is our default representation of *O*(Cˆ). Informally speaking, a *δ* ˆ -name of an open set

*O*(C)

is an exhaustive list of all basic neighborhoods whose closures are contained in that set. If this set is bounded, then this list will only contain bounded neighborhoods. Otherwise, it will contain both bounded and unbounded neighborhoods. These observations lead to the following.

**Proposition 3.4** *There is no computable F* :*⊆* Σ*ω → {*0*,* 1*} such that for all p ∈*

dom(*δO*(Cˆ))*,*

*F* (*p*) = ⎧⎨ 1 *if ∞∈ δO*(Cˆ)(*p*)

⎩ 0 *otherwise.*

We now turn to closed sets.

## Definition 3.5 (Computable topology on closed subsets of the extended plane)

* 1. For all *w ∈* dom(*ν*Cˆ ), let

*νC*(Cˆ)(*w*)= *{X ∈ C*(Cˆ) *| ν*Cˆ (*w*) *∩ X /*=*∅}*

Then, let *σC*(Cˆ) = ran(*νC*(Cˆ)).

* 1. Let *SC*(Cˆ) = (*C*(Cˆ)*, σC*(Cˆ)*, νC*(Cˆ)).

Hence, *SC*(Cˆ) is a computable topological space. Let *δC*(Cˆ) = *δSC*(Cˆ) . *δC*(Cˆ) is our default representation of *C*(Cˆ). Informally speaking, a *δ* ˆ -name of a closed set lists all basic neighborhoods that intersect the set.

*C*(C)

We now consider meromorphic functions. Let M(Cˆ) be the set of all meromor- phic *f* :*⊆* Cˆ *→* Cˆ.

**Definition 3.6** For all *f ∈* M(Cˆ) and all *p ∈* Σ*ω* , let *δ*M(Cˆ)(*p*) = *f* if there exist

*q, r* such that *p* = *⟨r, q⟩*, *ηωω* (*δ*ˆ*, δ*ˆ)-realizes *f* , and dom(*f* ) = *δ* ˆ (*r*).

*q* C C *O*(C)

*δ*M(Cˆ) is our default naming system for M(Cˆ). Informally speaking, a *δ*M(Cˆ)- name of a meromorphic function contains a name of the domain of the function as

well as an oracle Turing machine that computes the function. A thorough treatment of naming systems for function spaces is [[5](#_bookmark21)].

We have now defined our default naming systems for the spaces we will consider.

We now describe the default naming systems for the derived spaces.

1. **(Function spaces)** When *δX, δY* are the default naming systems for *X, Y* respectively, then [*δX → δY* ] is the default naming system for the set of all (*δX, δY* )-continuous functions with domain *X*. When, in addition, *A ⊆ X*, [*δX → δY* ]*A* is the default naming system for the set of (*δX, δY* )-continuous functions with domain *A*.
2. **(Products)** When *δX*1 *,..., δXn* are the default naming systems for *X*1*,..., Xn* respectively, then [*δX*1 *,..., δXn* ] is the default naming system for *X*1 *×...×Xn*. In addition, [*δX* ]*n* is the default naming system for *Xn* when *δX* is the default naming system for *X* and *n ≤ ω*.

When only the default naming systems are being used, we will suppress their mention. For example, we just say that a *δ*Cˆ -computable point in Cˆ is computable. This convention will eliminate much notation from our discussions. To eliminate

even more, let:

*zp* = *δ*Cˆ (*p*)

*Dp* = *δO*(Cˆ)(*p*)

*Ep* = *δC*(Cˆ)(*p*)

*fp* = *δ*M(Cˆ)(*p*)

* 1. *Some lemmas for proving computability*

Since *η*(*γ, z*) is integer-valued, we immediately obtain the following by using the fact that integration is a computable operator (see, for example, Theorem 6.4.1.2 of [[15](#_bookmark30)]).

**Proposition 3.7** *γ '→ Int*(*γ*) *and γ '→ Ext*(*γ*) *are computable.*

We now give some tools for proving computability without using type two ma- chines.

**Definition 3.8** A predicate *R ⊆* Σ*ω ×* Σ*∗* is *computable* if there is a type two machine *M* such that when *ι*(*w*)*p* is written on the input tape, *M* halts with output 1 if *R*(*p, w*) holds and halts with output 0 if *R*(*p, w*) does not hold. A predicate *S ⊆* (Σ*ω* )*n ×* (Σ*∗*)*m* is *computable* if there is a computable predicate *R ⊆* Σ*ω ×* Σ*∗* such that

*S*(*p*1*,..., pn, w*1*,..., wm*) *⇔ R*(*⟨p*1*,..., pn⟩, ⟨w*1*,..., wm⟩*)*.*

**Lemma 3.9** *Let S*1 = (*M*1*, τ*1*, ν*1) *be a computable topological space. Suppose* dom(*ν*1) *is computable. Let δ be a representation of M*0*. Let f* :*⊆ M*0 *→ M*1*.* *Then, the following are equivalent.*

* + 1. *f is* (*δ, δS*1 )*-computable.*
    2. *There is a computable predicate R ⊆* Σ*ω ×* Σ*∗ ×* Σ*∗ such that for all p ∈*

*δ—*1(dom(*f* )) *and all w ∈* dom(*ν*1)

*f* (*δ*(*p*)) *∈ ν*1(*w*) *⇔ EyR*(*p, w, y*)*.*

# Operations on points

We claim that the operations of addition, multiplication, and division are com- putable on Cˆ. We first review how these operations are defined on Cˆ and what their domains are.

**Definition 4.1** The operations of addition and multiplication are extended to Cˆ by the equations

*z* + *∞* = *∞* + *z* = *∞, z ∈* Cˆ

*z ×∞* = *∞× z* = *∞, z ∈* Cˆ *— {*0*}*

Division is extended to Cˆ by the equations

*z/∞* = 0*, z ∈* C

*∞/z* = *∞, z ∈* C

*z/*0= *∞, z ∈* Cˆ *— {*0*}*

Hence

dom(+) = Cˆ *×* Cˆ

dom(*×*)= Cˆ *×* Cˆ *— {*(0*, ∞*)*,* (*∞,* 0)*}*

dom(*/*)= Cˆ *×* Cˆ *— {*(*∞, ∞*)*,* (0*,* 0)*}*

The following can be proven fairly easily using Lemma [3.9](#_bookmark8), the definition of *δ*Cˆ , and Definition [4.1](#_bookmark9).

**Proposition 4.2** *Addition is computable on* Cˆ *×* Cˆ *— {*(*∞, ∞*)*}. Addition is not*

*continuous at* (*∞, ∞*)*. Hence, addition is not computable on* Cˆ *×* Cˆ*.*

**Proposition 4.3** *Multiplication of points in* Cˆ *is computable.*

**Proposition 4.4** *Division of points in* Cˆ *is computable.*

# Uniform computability of Mo¨bius transformations

Given distinct *z*1*, z*2*, z*3 *∈* Cˆ, there is a unique bilinear transformation that maps *z*1*, z*2*, z*3 to 1*,* 0*, ∞* respectively. Denote this transformation by *T*(*z*1*,z*2*,z*3). Details may be found in [[4](#_bookmark20)]. We claim the following.

**Theorem 5.1** *The map* (*z*1*, z*2*, z*3) *'→ T*(*z*1*,z*2*,z*3)*, where* (*z*1*, z*2*, z*3) *ranges over all*

*pairwise distinct triples in* Cˆ *×* Cˆ *×* Cˆ*, is computable.*

# Operations on meromorphic functions

We begin by extending some of the results in [[10](#_bookmark26)] and [[11](#_bookmark27)] on zeros to meromorphic functions.

**Lemma 6.1** *The following maps on* M(Cˆ) *are computable.*

1. *f '→* (Cˆ *— f—*1[*{*0*}*]) *∩* dom(*f* )*.*
2. *f '→* (Cˆ *— f—*1[*{∞}*]) *∩* dom(*f* )*.*

**Theorem 6.2** *The following maps are computable on* M(Cˆ)*.*

1. *f e f /≡*0 *'→ f—*1[*{*0*}*]*.*
2. *f e f /≡∞ →f —*1[*{∞}*]*.*

In order to prove that *f '→ f—*1 is computable, we must first prove some things about operations on open sets.

# Operations on closed and open sets

**Theorem 7.1** (i) *The map* (*f, U* ) *'→ f* [*U* ] *is computable on pairs such that f ∈*

M(Cˆ) *is not constant and U is an open subset of* dom(*f* )*.*

(ii) *The map* (*f, C*) *'→ f* [*C*] *is computable on pairs such that f ∈* M(Cˆ) *and C is a closed set that is contained in* dom(*f* )*.*

We can now show that computing inverses of injective meromorphic functions is possible.

**Theorem 7.2** *f '→ f—*1 *is computable on the set of injective meromorphic func- tions.*

# Decomposing boundaries

For all non-degenerate *D ∈ Ðn*(Cˆ), let *Compn*(*D*) consist of all (*C*1*,..., Cn*) such that *C*1*,..., Cn* are the connected components of the complement of *D*. Then, for all such *D*, let:

*Comp>*(*D*)= *{*(Cˆ *— C*1*,...,* Cˆ *— Cn*) *|* (*C*1*,..., Cn*) *∈ Compn*(*D*)*}*

*n*

*Comp∂* (*D*)= *{*(*∂C*1*,..., ∂Cn*) *|* (*C*1*,..., Cn*) *∈ Compn*(*D*)*}*

*n*

The proof of the following is based on Lemmas [2.10](#_bookmark3) and [2.11](#_bookmark5).

**Theorem 8.1** (*D, ∂D*) *'→ Comp>*(*D*) *and* (*D, ∂D*) *'→ Comp∂* (*D, ∂D*) *are com-*

*n n*

*putable multifunctions.*

# Constants associated with a circular domain

We claim that the constants associated with a circular domain can be computed from the domain and its boundary. These constants will be used in Section [10.1](#_bookmark13).

**Proposition 9.1** (*C, ∂C*) *'→ μC is computable (where C ranges over the circular regions in Ðn*(Cˆ) *for ﬁxed n).*

**Proposition 9.2** (*C, ∂C*) *'→ δC is computable (where C ranges over the circular regions in Ðn*(Cˆ) *for ﬁxed n).*

# Conformal mapping of *n*-connected domains

* 1. *Koebe’s algorithm*

The following can be proven using Hertling’s result on the Riemann Mapping The- orem [[7](#_bookmark23)] and the results of the previous sections.

**Theorem 10.1** *When D ranges over all non-degenerate domains in Ð*1(Cˆ) *that contain ∞ but not* 0*,* (*D, ∂D*) *'→* (*CD, ∂CD, fD*) *is computable.*

We will need the following for the proof of Theorem [10.3](#_bookmark16).

**Lemma 10.2** *From a δ*M(Cˆ)*-name of a map of the form f|*C *such that f ∈* M(Cˆ)

*and f has no poles except at ∞, it is possible to compute a name of f.*

**Theorem 10.3** *When D ranges over non-degenerate domains in Ðn*(Cˆ) *that con- tain ∞ but not* 0*,* (*D, ∂D, CD, ∂CD*) *'→ fD is computable.*

**Proof.** Given (*D, ∂D, CD, ∂CD*), we define sequences *{Dk}∞*

*k*=0

, *{Ck,*1*}∞*

, *.. .*,

*{Ck,n}∞*

*k*=0

*k*=0

, and *{fk}∞*

by simultaneous recursion as follows.

To begin, let *C*1*,..., Cn* be the connected components of Cˆ *— D*. We then let *D*0 = *D*. Let *C*0*,j* = *Cj* for *j* = 1*,..., n*. Let *f*0 = *Id*. By Theorem [8.1](#_bookmark11), we can compute names of *∂C*0*,j* and Cˆ *—* C0*,j* from names of *D, ∂D*.

*k*=0

Let *k ∈* N, and suppose *fk, Dk, Ck,*1*,..., Ck,n* have been defined. Assume we have also computed names of *fk*, *Dk*, and the complements and boundaries of

*Ck,*1*,..., Ck,n*. Let *k' ∈ {*1*,..., n}* be equivalent to *k* modulo *n*. Let *fk*+1 be the conformal map of Cˆ *—Ck,k'* onto a circular domain *C* such that *fk*+1(*z*) = *z*+*O*(*z—*1). Since *∂*(Cˆ *— Ck,k'* ) = *∂Ck,k'* , by Theorem [10.1](#_bookmark14), we can compute a name of *fk*+1.

Now, let *Dk*+1 = *fk*+1[*Dk*]. Let *Ck*+1*,j* = *fk*+1[*Ck,j*] when *j /*=*k'*. Let *Ck*+1*,k'* = Cˆ *— C*. Note that when *j /*=*k'*, *Ck,j ⊆* dom(*fk*+1) and so we can compute a name of *∂Ck*+1*,j*. By Theorem [10.1](#_bookmark14), we can compute a name of *∂Ck*+1*,k'* . At the same time, we note that Cˆ *— Ck*+1*,j* = *Ck*+1*,k' ∪ fk*+1[Cˆ *— Ck,j*] when *j /*=*k'*. Hence, we can compute names of the complements of *Ck*+1*,*1*,..., Ck*+1*,n*.

It follows that the sequence *{fk}∞* can be computed from *D* and *∂D*. (See,

*k*=0

for example, Theorem 2.1.14 of [[15](#_bookmark30)].) We now let:

*g*0 = *f*1

*gk*+1 = *fk*+2 *◦ gk*

The following is proven in [[6](#_bookmark22)].

**Lemma 10.4** *The sequence {gk}∞*

*k*=0

*converges pointwise to a conformal map g of*

*D onto CD such that g*(*z*) = *z* + *O*(*z—*1)*.*

So *fD* = *g*.

Choose *ρ* so that *ρ >* max*{|z|* : *z ∈* Cˆ *— CD}*. Since this maximum occurs on

*∂CD*, *ρ* can be computed from the given information.

Abbreviate *CD* with *C*. Let *μ* = *μC* and *δ* = *δC*. Let

2*ρ*2 2[*πμ—*1]2

*γ* =

+ 1 *.*

*πδ* ln *μ—*1

The following is proven in [[6](#_bookmark22)].

**Lemma 10.5** *For all z ∈ D — {∞} and all j ∈* N*, |gj*(*z*) *— g*(*z*)*|≤ γμ*4*[j/n♩.*

It now follows from Theorem 6.2.2.2 of [[15](#_bookmark30)] that we can compute a [*ρ*2 *→ ρ*2]- name of the restriction of *g* to *D ∩* C. Then, by Lemma [10.2](#_bookmark15) we can compute a name of *g*. This completes the proof.

From the proof of Theorem [10.3](#_bookmark16), we can extract a proof of the following non- uniform result.

**Theorem 10.6** *Let D ∈ Ðn*(Cˆ) *be non-degenerate. If D, ∂D are computable, then*

*CD, fD are computable.*

**Proof.** Choose rational *R, r* such that *R > γ* and *μ < r <* 1.

* 1. *Necessity of parameters*

Suppose we have a (possibly non-computable) function *x '→ f* (*x*). A *sufficient parameter* for this function is a function *g* such that (*x, g*(*x*)) *'→ f* (*x*) is computable. If *x '→ g*(*x*) is computable, then we call *g superfluous* (and it follows that *f* is

computable). If *g* is sufficient and (*x, f*(*x*)) *'→ g*(*x*) is computable, then we call the parameter *g exact*. The parameter *g* is called *necessary* if it is both exact and non-superfluous.

This terminology can be extended to situations where we add several parameters. For example, suppose we add two parameters *g*1*, g*2 so that (*x, g*1(*x*)*, g*2(*x*)) *'→ f* (*x*) is computable. We say that *g*1 is *exact relative to g*2 if (*x, f* (*x*)*, g*2(*x*)) *'→ g*1(*x*) is computable. We then say that *g*1 is *superfluous relative to g*2 if (*x, g*2(*x*)) *'→ g*1(*x*) is computable. We say that *g*1 is *necessary relative to g*2 if it is exact and non- superfluous relative to *g*2. We similarly define these terms for *g*2 relative to *g*1. We then extend this terminology to the addition of *n >* 2 parameters in the obvious way.

We now claim that with respect to the map *D '→ fD*, the parameter *∂D* is exact with respect to the additional parameters (*CD, ∂CD*). The proof is similar to the proof of Theorem 4.7 of [[7](#_bookmark23)].

**Theorem 10.7** *The map* (*D, fD, CD, ∂CD*) *'→ ∂D, where D ranges over non- degenerate domains in Ðn*(Cˆ) *that contain ∞ but not* 0 *is computable.*

We can similarly show that *∂CD* is exact with respect to (*∂D, CD*). In fact, we can show that (*∂D, fD*) *'→ ∂CD* is computable. We can also show that *CD* is exact with respect to (*∂D, ∂CD*). In fact, it just follows from Theorem [7.1](#_bookmark10) that *fD '→ CD* is computable. We do not know if these parameters are necessary.

# References

1. E. Bishop and D.S. Bridges, *Constructive analysis*, Springer (Berlin), 1985.
2. H. Cheng, *A constructive Riemann mapping theorem*, Pacific Journal Mathematics **44** (1973), 435 – 454.
3. D. Gaier, *Untersuchung zur durchfu¨hrung der konformen abbildung mehrfach zusammenha¨ngender gebiete*, Arch. Rat. Mech. Anal. **3** (1959), 149 – 178.
4. T. W. Gamelin, *Complex analysis*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 2001.
5. T. Grubba, K. Weihrauch, and Y. Xu, *Effectivity on continuous functions in topological spaces*, Electronic Notes in Theoretical Computer Science **202** (2008), 237–254.
6. P. Henrici, *Applied and computational complex analysis. vol. 3.*, Pure and Applied Mathematics, Wiley & Sons, Inc., New York, 1986.
7. P. Hertling, *An effective Riemann Mapping Theorem*, Theoretical Computer Science **219** (1999), 225 – 265.
8. P. Koebe, *U¨ber die konforme abbildung mehrfach-zusammenha¨ngender bereiche*, Jahresber. Dt. Math. Ver. **19** (1910), 339 – 348.
9. , *U¨ber eine neue methode der konformen abbildung und uniformisierung*, Nachr. Kgl. Ges. Wiss. G¨ottingen, Math.-Phys. Kl. 1912 (1912), 844–848.
10. A. Matheson and T.H. McNicholl, *Computable analysis and Blaschke products*, Proceedings of the American Mathematical Society **1** (2008), 321 – 332 (electronic).
11. T.H. McNicholl, *Uniformly computable aspects of inner functions: estimation and factorization*, Mathematical Logic Quarterly (to appear).
12. Z. Nehari, *Conformal mappings*, Dover, 1975.
13. R. Rettinger, *A fast algorithm for julia sets of hyperbolic rational functions*, Electronic Notes in Theoretical Computer Science **120** (2005), 145 – 157.
14. M. Tsuji, *Potential theory*, Chelsea, New York, 1975.
15. K. Weihrauch, *Computable analysis, an introduction*, Springer-Verlag, 2000.