

COMPUTATION OF CAPACITY

THOMAS RANSFORD AND JÉRÉMIE ROSTAND

ABSTRACT. This article introduces a method for computing upper and lower bounds for the logarithmic capacity of a compact plane set. If the set has the Hölder continuity property, then these bounds converge to the value of the capacity. A number of examples are discussed in detail, including the Cantor middle-third set, for which we estimate $c(E) \approx 0.220949102189507$.

1. INTRODUCTION

The capacity of a set in \mathbb{R}^n measures its size from the point of view of potential theory. In particular, the sets of capacity zero play the role of negligible sets in potential theory in much the same way that sets of measure zero are negligible in measure theory. But there are also occasions when the actual value of the capacity is important. This is particularly true of the logarithmic capacity in \mathbb{R}^2 , which is the case that we study in this article.

It has been known since the early 1930's, thanks to the work of Fekete and Szegő, that logarithmic capacity is closely linked with complex analysis, where it appears in several other guises. For example, it is the same thing as the Chebyshev constant, which plays a key role in polynomial approximation [15]. It is also equal to the transfinite diameter, where it leads to several applications in number theory [3]. Also, as the (negative exponential of the) Robin constant, it is intimately connected with Green's functions and conformal mapping [10, 12, 13]. This latter connection can be exploited to compute the capacity in a few simple cases. For instance, the capacity of a disk is equal to its radius, and the capacity of a line segment is one quarter of its length. For a list of other examples, see e.g. [13, p.135].

In general, however, capacity is notoriously hard to compute. Unless the set in question is connected, or has some special symmetry, the conformal mapping techniques break down. The purpose of this article is to describe a numerical method which provides rigorous upper and lower bounds for the capacity of any compact plane set. If the set has the Hölder continuity property, then these bounds can in principle be made arbitrarily close to one another, and therefore converge to the true value of the capacity. We shall give several examples to illustrate how the method works in practice, in particular computing the capacity of the Cantor middle-third set.

Received by the editor January 18, 2005 and, in revised form, July 6, 2005.

2000 *Mathematics Subject Classification.* Primary 65E05; Secondary 31A15, 90C05.

Key words and phrases. Capacity, matrix game, Hölder continuity property, Cantor set.

The first author was supported by grants from NSERC and the Canada Research Chairs program.

The second author was supported by a grant from NSERC.

The paper is organized as follows. After establishing basic definitions and notation in §2, we develop in §3 the theory that leads to the bounds for capacity and their convergence to the true value. In §4 we describe the numerical methods used, and then present a number of illustrative examples in §5. Finally, in §6, we apply our techniques to investigate the variation of the capacity of a family of generalized Cantor sets.

2. DEFINITIONS AND NOTATION

2.1. Capacity. Let E be a compact subset of \mathbb{C} . We write $\mathcal{P}(E)$ for the family of all Borel probability measures on E . Given $\mu \in \mathcal{P}(E)$, we define its *energy* by

$$I(\mu) := \iint \log \left| \frac{1}{z-w} \right| d\mu(z) d\mu(w).$$

If there exists $\mu \in \mathcal{P}(E)$ such that $I(\mu) < \infty$, then there exists a unique $\nu \in \mathcal{P}(E)$, called the *equilibrium measure*, such that

$$I(\nu) = \min_{\mu \in \mathcal{P}(E)} I(\mu).$$

The *capacity* of E is then defined by

$$c(E) := e^{-I(\nu)}.$$

In the remaining case, namely when $I(\mu) = \infty$ for all $\mu \in \mathcal{P}(E)$, we define $c(E) := 0$. For more information about capacity, see e.g. [13, §5].

2.2. Matrix games. Our method for computing capacity will be to convert the problem to the calculation of the value of certain matrix games. We now briefly introduce the notation used.

Fix a positive integer n , and set $\Delta_n := \{(t_1, \dots, t_n) : t_j \geq 0, \sum_j t_j = 1\}$. Given an $n \times n$ matrix $h = (h_{ij})$, we shall write

$$M(h) := \min_{s \in \Delta_n} \max_{t \in \Delta_n} \sum_{i,j} h_{ij} s_i t_j = \max_{t \in \Delta_n} \min_{s \in \Delta_n} \sum_{i,j} h_{ij} s_i t_j.$$

The equality of the two quantities defining $M(h)$ is a simple consequence of the von Neumann minimax principle [16, Theorem 1.3]. Notice that, since a convex combination of numbers always lies between their maximum and minimum, we also have

$$M(h) = \min_{s \in \Delta_n} \max_j \sum_i h_{ij} s_i = \max_{t \in \Delta_n} \min_i \sum_j h_{ij} t_j.$$

In practice $M(h)$ can be computed as the solution to a linear programming problem. We shall return to this in §4.1.

2.3. Other notation. Throughout the paper $B(w, r)$ will denote the closed disk with center w and radius r . Also, given compact subsets E, F of \mathbb{C} , we shall write $\text{diam}(E)$ for the diameter of E , and $\text{dist}(E, F)$ for the distance between E and F .

3. BOUNDS FOR CAPACITY AND THE MAIN CONVERGENCE THEOREM

In this section, we consider the following general set-up. Let E be the compact subset of \mathbb{C} whose capacity we would like to compute. We suppose that we have compact subsets F_1, \dots, F_n of \mathbb{C} and a real number $\delta > 0$ such that

$$(3.1) \quad \begin{cases} E \subset F_1 \cup \dots \cup F_n, \\ c(E \cap F_i) \geq \delta \quad (i = 1, \dots, n). \end{cases}$$

We define symmetric $n \times n$ matrices a and b by

$$(3.2) \quad a_{ij} := \log \frac{1}{\text{diam}(F_i \cup F_j)}, \quad b_{ij} := \log \frac{1}{\max(\delta, \text{dist}(F_i, F_j))}.$$

3.1. Bounds for the capacity. Our method for computing the capacity of E is based upon the following theorem, which provides upper and lower bounds for the capacity in terms of $M(a)$ and $M(b)$.

Theorem 3.1. *Let E be a compact subset of \mathbb{C} . Suppose that F_1, \dots, F_n, δ satisfy (3.1) and that a, b are defined as in (3.2). Then*

$$M(a) \leq \log \frac{1}{c(E)} \leq M(b).$$

This theorem will be proved in §3.3 below.

At first sight, the theorem appears to imply a circular argument, since, to estimate the capacity of E , we first need to choose F_1, \dots, F_n with $c(E \cap F_i) \geq \delta$, and so we need to estimate these capacities as well. However, as we shall see, even fairly rough estimates for $c(E \cap F_i)$ lead to much better estimates for $c(E)$.

This still leaves us with the problem of finding initial estimates for $c(E \cap F_i)$. There are several approaches to this.

The simplest case is when E is connected. In this situation, it follows using [13, Theorem 5.3.3 (a)] that

$$c(E \cap B(w, r)) \geq r/4 \quad (w \in E, 0 < r \leq \text{diam}(E)).$$

Thus we can take a covering of E of the form $F_i := B(w_i, r)$ ($w_i \in E$), and $\delta := r/4$.

A more general situation is when E is *uniformly perfect*. This means that there exists a constant $\beta > 0$ such that

$$(3.3) \quad E \cap \{z : \beta r \leq |z - w| \leq r\} \neq \emptyset \quad (w \in E, 0 < r \leq \text{diam}(E)).$$

It is easy to see that if E is connected (and has more than one point), then it is uniformly perfect. A finite union of uniformly perfect sets is again uniformly perfect. There are also interesting examples of uniformly perfect sets with infinitely many components, such as Julia sets of rational functions [4, Theorem 3.3], and limit sets of certain iterated function systems [1, Corollary 1.4], for instance the Cantor middle-third set. A fundamental fact about uniformly perfect sets is the following inequality, due to Pommerenke [11]: if E satisfies (3.3), then

$$c(E \cap B(w, r)) \geq \beta^2 r/32 \quad (w \in E, 0 < r \leq \text{diam}(E)).$$

Thus in this case, we can again take $F_i := B(w_i, r)$, now with $\delta := \beta^2 r/32$.

A third option, which is sometimes available when E exhibits affine self-similarity, is to choose the F_i so that each set $E \cap F_i$ is just a scaled copy of E , say $E \cap F_i = \lambda_i E + \tau_i$, where $\lambda_i, \tau_i \in \mathbb{C}$. In that case, $c(E \cap F_i) = |\lambda_i|c(E)$, and so if we already

have a lower bound c_0 for $c(E)$ (either *a priori*, or via a previous application of Theorem 3.1), then we can take $\delta = c_0 \min_i |\lambda_i|$.

3.2. Convergence. Theorem 3.1 yields upper and lower bounds for $c(E)$, but it says nothing about how close together they are. This is the subject of our second theorem. To state it, we need some definitions.

We shall say that a continuous function $g_E : \mathbb{C} \rightarrow \mathbb{R}$ is the *Green's function* for E if

$$\begin{cases} g_E \text{ is harmonic on } \mathbb{C} \setminus E, \\ g_E = 0 \text{ on } E, \\ g_E(z) = \log |z| + O(1) \text{ as } |z| \rightarrow \infty. \end{cases}$$

These conditions determine g_E uniquely, if it exists. (Strictly speaking, what we have defined is the Green's function of the unbounded component of $\mathbb{C}_\infty \setminus E$ with pole at infinity, extended to be zero elsewhere in \mathbb{C} . For more information on Green's functions see e.g. [13, §4.4].)

A compact set E is said to have the *Hölder continuity property (HCP)* if its Green's function g_E exists and satisfies

$$|g_E(z) - g_E(w)| \leq A|z - w|^\alpha \quad (z, w \in \mathbb{C})$$

for some constants $A, \alpha > 0$. It is known that every uniformly perfect set has HCP, and that if a set E is connected or has just finitely many components (none of them singletons), then in fact it satisfies HCP with exponent $\alpha = 1/2$. For more details about HCP, see for instance the recent paper [5].

Theorem 3.2. *Let E be a compact set which satisfies HCP with exponent $\alpha > 0$. Suppose that F_1, \dots, F_n, δ satisfy (3.1) and that a, b are defined as in (3.2). Then*

$$(3.4) \quad M(b) - M(a) \leq C_E d^{\alpha/(\alpha+1)} \log \left(\frac{\text{diam}(E)}{\delta} \right),$$

where $d := \max_i \text{diam}(F_i)$, and C_E is a constant depending only on E .

The theorem will be proved in §3.3 below.

It demonstrates that, if we can choose coverings (F_i) of E such that δ is proportional to $d := \max_i \text{diam}(F_i)$, then $M(b) - M(a) = O(d^{\alpha/(\alpha+1)} \log(1/d))$ as $d \rightarrow 0$, and therefore $M(a), M(b) \rightarrow \log 1/c(E)$ as $d \rightarrow 0$. From what we said above, this will always be the case if E is uniformly perfect. If, further, E has only finitely many components, then we may take $\alpha = 1/2$, and so in this case the error is $O(d^{1/3} \log(1/d))$ as $d \rightarrow 0$.

We remark that the computational effort involved in calculating $M(a)$ and $M(b)$ depends directly on n , the number of sets F_j used to cover E . To cover E by n sets of diameter d , we need to take n to be of the order $d^{-\dim(E)}$, where $\dim(E)$ denotes the Minkowski dimension of E (see [8, §3.1]). Thus, from this point of view at least, sets of smaller dimension are better. In particular, if E has non-empty interior, then one should profit from the fact that $c(E) = c(\partial E)$ (see e.g. [13, Theorem 5.1.2]), and compute $c(\partial E)$ instead.

3.3. Proofs of the theorems. We begin with the proof of Theorem 3.1. Let E be a compact subset of \mathbb{C} . We suppose that F_1, \dots, F_n, δ satisfy (3.1) and that the matrices a, b are defined as in (3.2).

Lemma 3.3. *Let G_1, \dots, G_n be a Borel partition of E such that $G_j \subset F_j$ for all j . Let ν be the equilibrium measure for E . Then, for each $i \in \{1, \dots, n\}$,*

$$\sum_j a_{ij} \nu(G_j) \leq \log \frac{1}{c(E)} \leq \sum_j b_{ij} \nu(G_j).$$

Proof. Fix $i \in \{1, \dots, n\}$. Let ν_i be the equilibrium measure of $E \cap F_i$. Then its potential

$$U_{\nu_i}(z) := \int \log \left| \frac{1}{z-w} \right| d\nu_i(w)$$

clearly satisfies

$$\log \frac{1}{\text{diam}(F_i \cup F_j)} \leq U_{\nu_i}(z) \leq \log \frac{1}{\text{dist}(F_i, F_j)} \quad (z \in F_j).$$

Also, by Frostman's theorem [13, Theorem 3.3.4],

$$U_{\nu_i}(z) \leq I(\nu_i) = \log \frac{1}{c(E \cap F_i)} \leq \log \frac{1}{\delta} \quad (z \in \mathbb{C}).$$

Putting these together we deduce that

$$a_{ij} \leq U_{\nu_i}(z) \leq b_{ij} \quad (z \in F_j),$$

where a_{ij}, b_{ij} are defined as in (3.2).

Now, by Frostman's theorem again, $U_\nu = I(\nu) = \log 1/c(E)$ on E , outside an exceptional set of capacity zero, and hence of ν_i -measure zero. Using Fubini's theorem, it follows that

$$\log \frac{1}{c(E)} = \int U_\nu d\nu = \int U_{\nu_i} d\nu = \sum_j \int_{G_j} U_{\nu_i} d\nu.$$

As $a_{ij} \leq U_{\nu_i} \leq b_{ij}$ on G_j , this last sum lies between $\sum_j a_{ij} \nu(G_j)$ and $\sum_j b_{ij} \nu(G_j)$. This completes the proof. \square

Completion of the proof of Theorem 3.1. Choose G_1, \dots, G_n as in the statement of the lemma; for example, take $G_j := E \cap F_j \setminus (F_1 \cup \dots \cup F_{j-1})$. Then, by the lemma,

$$\log \frac{1}{c(E)} \leq \min_i \sum_j b_{ij} \nu(G_j) \leq \max_{t \in \Delta_n} \min_i \sum_j b_{ij} t_j = M(b).$$

Likewise,

$$\log \frac{1}{c(E)} \geq \max_i \sum_j a_{ij} \nu(G_j) \geq \min_{s \in \Delta_n} \max_i \sum_j a_{ij} s_j = M(a),$$

where for the last equality we used the symmetry of a . This completes the proof. \square

We now turn to the proof of Theorem 3.2. Let E be a compact subset of \mathbb{C} . Once again we suppose that F_1, \dots, F_n, δ satisfy (3.1) and that the matrices a, b are defined as in (3.2). Also, we set $d := \max_j \text{diam}(F_j)$.

Lemma 3.4. *Define $\omega(r) := \sup_{w \in E} \nu(B(w, r))$, where ν is the equilibrium measure for E . Then*

$$M(b) - M(a) \leq \inf_{r \geq 4d} \left(2\omega(r) \log \frac{r}{\delta} + \frac{8d}{r} \right).$$

Proof. We re-use Lemma 3.3. Choose G_1, \dots, G_n as in that lemma. Then

$$\log \frac{1}{c(E)} - M(a) \leq \min_i \sum_j b_{ij} \nu(G_j) - \min_i \sum_j a_{ij} \nu(G_j) \leq \max_i \sum_j (b_{ij} - a_{ij}) \nu(G_j).$$

Likewise,

$$M(b) - \log \frac{1}{c(E)} \leq \max_i \sum_j b_{ij} \nu(G_j) - \max_i \sum_j a_{ij} \nu(G_j) \leq \max_i \sum_j (b_{ij} - a_{ij}) \nu(G_j).$$

Adding, we obtain

$$(3.5) \quad M(b) - M(a) \leq 2 \max_i \sum_j (b_{ij} - a_{ij}) \nu(G_j).$$

Now fix $i \in \{1, \dots, n\}$. We shall estimate the sum $\sum_j (b_{ij} - a_{ij}) \nu(G_j)$ by splitting it into two. Let $r \geq 4d$, and define

$$J_r := \{j : \text{diam}(F_i \cup F_j) \leq r\} \quad \text{and} \quad J'_r := \{j : \text{diam}(F_i \cup F_j) > r\}.$$

If $j \in J_r$, then $a_{ij} \geq \log(1/r)$ and $b_{ij} \leq \log(1/\delta)$. Also the sets G_j ($j \in J_r$) are disjoint subsets of $B(w, r)$, for some $w \in E \cap F_i$. Hence

$$\sum_{j \in J_r} \nu(G_j)(b_{ij} - a_{ij}) \leq \sum_{j \in J_r} \nu(G_j) \log \frac{r}{\delta} = \nu\left(\bigcup_{j \in J_r} G_j\right) \log \frac{r}{\delta} \leq \omega(r) \log \frac{r}{\delta}.$$

If $j \in J'_r$, then $\text{diam}(F_i \cup F_j) > r$, and since $\text{dist}(F_i, F_j) \geq \text{diam}(F_i \cup F_j) - 2d$, we have

$$b_{ij} - a_{ij} \leq \log \frac{\text{diam}(F_i \cup F_j)}{\text{dist}(F_i, F_j)} \leq \log \frac{\text{diam}(F_i \cup F_j)}{\text{diam}(F_i \cup F_j) - 2d} \leq \log \frac{r}{r - 2d} \leq \frac{4d}{r},$$

where for the last inequality we have used the facts that $\log x \leq x - 1$ and $r \geq 4d$. Hence

$$\sum_{j \in J'_r} \nu(G_j)(b_{ij} - a_{ij}) \leq \sum_{j \in J'_r} \nu(G_j) \frac{4d}{r} = \nu\left(\bigcup_{j \in J'_r} G_j\right) \frac{4d}{r} \leq \frac{4d}{r}.$$

Putting these estimates together, and noting that they hold for all $r \geq 4d$, we obtain

$$(3.6) \quad \sum_j \nu(G_j)(b_{ij} - a_{ij}) \leq \inf_{r \geq 4d} \left(\omega(r) \log \frac{r}{\delta} + \frac{4d}{r} \right).$$

The result follows upon combining (3.5) and (3.6). \square

To apply this lemma, we need to be able to estimate $\nu(B(w, r))$, where ν is the equilibrium measure of E . This can be done via the Green's function of E .

Lemma 3.5. *Suppose that E has equilibrium measure ν and Green's function g_E . Then*

$$\nu(B(w, r)) \leq \max_{|z-w|=er} (g_E(z) - g_E(w)) \quad (w \in \mathbb{C}, r > 0).$$

Proof. Our starting point is the following formula relating the Green's function to the equilibrium measure (see e.g. [13, p.107]):

$$g_E(\zeta) = \int_E \log |\zeta - z| d\nu(z) - \log c(E) \quad (\zeta \in \mathbb{C}).$$

It follows that, for $w \in \mathbb{C}$ and $r > 0$,

$$\begin{aligned}
& \frac{1}{2\pi} \int_0^{2\pi} (g_E(w + re^{i\theta}) - g_E(w)) d\theta \\
&= \frac{1}{2\pi} \int_0^{2\pi} \int_E \left(\log |w + re^{i\theta} - z| - \log |w - z| \right) d\nu(z) d\theta \\
&= \int_E \left(\frac{1}{2\pi} \int_0^{2\pi} \log \left| 1 + \frac{re^{i\theta}}{w - z} \right| d\theta \right) d\nu(z) \\
&= \int_E \log^+ \left| \frac{r}{w - z} \right| d\nu(z) \\
&\geq \nu(B(w, r/e)).
\end{aligned}$$

The result follows upon replacing r by er throughout. \square

Completion of the proof of Theorem 3.2. We assume, as in the statement of the theorem, that E satisfies the HCP with exponent α . In other words, the Green's function g_E exists and satisfies

$$|g_E(z) - g_E(w)| \leq A|z - w|^\alpha \quad (z, w \in \mathbb{C}).$$

By Lemma 3.5, the equilibrium measure ν of E satisfies $\nu(B(w, r)) \leq Ae^\alpha r^\alpha$ for all $w \in \mathbb{C}$ and $r > 0$. Hence, by Lemma 3.4,

$$M(b) - M(a) \leq \inf_{r \geq 4d} \left(2Ae^\alpha r^\alpha \log \frac{r}{\delta} + \frac{8d}{r} \right).$$

Suppose for the moment that $d \leq \text{diam}(E)$. Set $r := 4d^{1/(\alpha+1)} \text{diam}(E)^{\alpha/(\alpha+1)}$. Then $r \geq 4d$, and substituting it in the inequality above, we obtain

$$M(b) - M(a) \leq C_1 d^{\alpha/(\alpha+1)} \log \left(\frac{4 \text{diam}(E)}{\delta} \right) + C_2 d^{\alpha/(\alpha+1)},$$

where C_1, C_2 are constants depending just on A, α and $\text{diam}(E)$. Since the capacity of a set is never more than half the diameter (see [13, Theorem 5.3.4]), we have $\text{diam}(E) \geq 2\delta$. Thus the second term on the right-hand side above can be absorbed into the first, and the factor of 4 inside the log eliminated, adjusting C_1 appropriately. This gives (3.4), thereby proving the theorem in the case $d \leq \text{diam}(E)$.

The remaining case $d \geq \text{diam}(E)$ is of less importance, since we are mainly interested in what happens as $d \rightarrow 0$, but we sketch a brief proof for the sake of completeness. In this case we just use the estimates $a_{ij} \geq \log 1/(\text{diam}(E) + 2d)$ and $b_{ij} \leq \log(1/\delta)$ that arise directly from the definitions of a, b , to obtain

$$M(b) - M(a) \leq \log \left(\frac{\text{diam}(E) + 2d}{\delta} \right).$$

Together with the fact that $d \geq \text{diam}(E) \geq 2\delta$, this eventually leads to (3.4). \square

4. METHODS OF COMPUTATION

We now discuss the practical implementation of the ideas of the previous section. In what follows, a vector or a matrix with scalar c in all entries will be simply denoted by c . It should be clear from the context what the symbol refers to. Also, it will be convenient to write $x \geq y$ to indicate that all entries of $x - y$ are non-negative, x and y being vectors or matrices of the same dimensions.

4.1. Computation of $M(h)$. Our method depends on being able to compute $M(h)$ for symmetric $n \times n$ matrices h . Here we describe several methods of doing this.

4.1.1. Reformulation via linear programming. As stated in §2.2, we have

$$(4.1) \quad M(h) := \min_{t \in \Delta_n} \max_j \sum_i h_{ij} t_i,$$

which is a convex non-linear programming problem. We can turn it into a linear programming problem by use of an auxiliary variable T . Consider the following problem:

$$(4.2) \quad \begin{cases} \text{minimize } T \text{ subject to the constraints} \\ t \geq 0, \quad \sum_j t_j = 1, \quad \sum_i h_{ij} t_i \leq T \quad (j = 1, \dots, n). \end{cases}$$

It is well known that $M(h)$ is equal to the value of T at the optimal point (see e.g. [16, §1.13]).

4.1.2. Linear system. Solving large dense linear programs is quite time-consuming, much more so than solving linear systems for instance. In some circumstances, $M(h)$ can be obtained by solving a linear equation instead of a linear program, as the following simple result shows.

Proposition 4.1. *If h is invertible, if $s := h^{-1}1 \geq 0$ and if $S := \sum_j s_j > 0$, then $t^* := S^{-1}s$ is the solution of (4.1), and $M(h) = S^{-1}$.*

Proof. On the one hand, it is easy to see that $t^* \in \Delta_n$, and since

$$\max_j \sum_i h_{ij} t_i^* = \max_j S^{-1} = S^{-1},$$

we have $M(h) \leq S^{-1}$. On the other hand, for all $t \in \Delta_n$ we have

$$\max_j \sum_i h_{ij} t_i \geq \sum_j \left(\sum_i h_{ij} t_i \right) t_j^* = \sum_i \left(\sum_j h_{ij} t_j^* \right) t_i = \sum_i S^{-1} t_i = S^{-1},$$

and therefore $M(h) \geq S^{-1}$. \square

4.1.3. Scaling. Sometimes, even if $h^{-1}1 \not\geq 0$, it may be possible to find a constant $\gamma > 0$ such that $\tilde{h}^{-1}1 \geq 0$, where \tilde{h} is the matrix associated with the compact set γE . Since $c(\gamma E) = \gamma c(E)$, we get that $\gamma^{-1}e^{-M(\tilde{h})}$ is an estimate for $c(E)$.

4.1.4. Other ways to bypass the linear program. There may be other options that run faster than a linear program, though at the cost of yielding worse bounds.

Upper bounds. For all $t \in \Delta_n$, we have $\max_j \sum_i t_i h_{ij} \geq \sum_{i,j} t_i t_j h_{ij}$. Therefore $M(h)$ is bounded below by a quadratic program, namely

$$Q(h) := \min_{t \in \Delta_n} \sum_{i,j} t_i t_j h_{ij}.$$

Consequently, $e^{-Q(h)}$ is an upper bound for $c(E)$. In practice, $Q(h)$ is often sufficiently close to $M(h)$ to give a useful upper bound. This approach is interesting as long as the quadratic program can be solved faster than the linear program. This frequently proves to be the case, though not always.

Lower bounds. At the other end, $M(h)$ is bounded above by $M_t(h) := \max_j \sum_i h_{ij} t_i$ for each feasible vector t . Therefore $e^{-M_t(h)}$ is a lower bound for $c(E)$ for every $t \in \Delta_n$. There are several ways to build a vector t that will give a useful bound.

If the quadratic program runs faster than the linear program, then we can use its solution t . The situation here is a bit different from that above. Indeed, even though $Q(h)$ may happen to be close to $M(h)$, the distance between $M_t(h)$ and $M(h)$ will generally be bigger.

Another possibility is to construct an educated guess for t , based on the solution obtained for a previous value of n . This is highly dependent on the geometry of E and on the choices of F_i and n .

Let us mention a third approach that exploits some ideas of Mehrotra's celebrated paper [9]. The points on the hyperplane $\sum_j t_j = 1$ that satisfy $t \geq 0$ are those that are closest to the origin. Thus it is natural to search for vectors t that have Euclidean norms as small as possible. One search direction is prompted by Proposition 4.1. Under the hypotheses of this proposition, at the optimal point we have $\sum_i h_{ij} t_i^* = S^{-1}$ for all i . This suggests looking for a scalar c and a vector r of small Euclidean norm $\|r\|$ such that $ht + r = c$. Since, in this context, c is an approximation to $M(h)$, we want to minimize its value as well. Combining all these ideas, we are left with the least-square problem of minimizing $\|x\|$ subject to the linear constraint $Ax = u$, where

$$A := \begin{pmatrix} h & I & -\lambda \\ 1 & 0 & 0 \end{pmatrix}_{(n+1) \times (n+n+1)}, \quad x := \begin{pmatrix} t \\ r \\ c/\lambda \end{pmatrix} \quad \text{and} \quad u := \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

and where $\lambda > 0$ is a real parameter to be chosen. If the t -component of the solution satisfies $t \geq 0$, then we have a potentially good feasible t . In practice, the solution to the least-square problem is calculated via the formula $A^t(AA^t)^{-1}u$, which is much faster than solving a linear program. Even if t is not positive, then in general it is very close to being so, and we can apply some heuristics to modify certain components to get a feasible t in the neighborhood. In the matrix formulation above we used c/λ instead of c , because this special component of x may not have the same order of magnitude as the others and we may want to give it a special weight in the computation of $\|x\|$. As a matter of fact, experimenting with different values of λ can lead to better bounds.

4.1.5. Symmetry. Finally, another trick sometimes available is to exploit the symmetry of the compact set under consideration. If h is built so that it is symmetric with respect to both its main diagonal and anti-diagonal, then the optimal vector t , at which we get $M(h)$, is also symmetric, i.e. $t_i = t_{n-i+1}$ for all i . This enables us to reduce the size of the linear program or system by replacing h by $(h_{ij} + h_{n-i+1,j})_{1 \leq i,j \leq n/2}$ and the condition on t by $t_1 + \dots + t_{n/2} = 1/2$. As a consequence, with the same storage and computation time we can solve the problem with twice as many F_i . Note that, in order to apply this reduction scheme, we have to label the F_i carefully. In some circumstances, more complex symmetries of E can be exploited to reduce the problem size still further.

4.2. Intermediate methods. Rather than calculating the matrices a and b directly, it may well be easier to compute some other symmetric matrix h satisfying $a \leq h \leq b$. Then we have $M(a) \leq M(h) \leq M(b)$, and so the fact that $M(a), M(b)$ converge to $\log 1/c(E)$ forces $M(h)$ to converge to the same limit, and at least as

quickly. In practice, the convergence may well be quicker. In this paragraph we present several ways to build such intermediate matrices h .

The effectiveness of the general methods presented below are quite dependent on the geometry of E and the choice of the F_i . One should keep in mind that, for a particular set E , there may be natural ways to construct h that are more closely related to E and that are more effective.

4.2.1. Midpoint (MP). For each i , pick a point $x_i \in F_i$ such that $x_i \neq x_j$ for all $i \neq j$. Typically, if F_i is an interval or a square, we choose for x_i its midpoint. Now let us define $h_{ij} := \log 1/\max(\delta, |x_i - x_j|)$. For all i, j we have

$$(4.3) \quad \text{dist}(F_i, F_j) \leq |x_i - x_j| \leq \text{diam}(F_i \cup F_j).$$

Since, for all i, j ,

$$\text{diam}(F_i \cup F_j) \geq \text{diam}(F_i) \geq c(F_i) \geq c(F_i \cap E) \geq \delta,$$

we get, upon taking the maximum with δ in (4.3), that $a \leq h \leq b$. Thus $e^{-M(h)}$ gives an approximation to $c(E)$.

4.2.2. Weighted average 1 (WA1). Choose $\alpha \in (0, 1)$ and set $h := \alpha a + (1 - \alpha)b$. A natural try is $\alpha = \frac{1}{2}$. Another possibility is based on the following reasoning. For increasing values of n , the sequences $M(a)$ and $M(b)$ converges to $\log 1/c(E)$ from below and above respectively. In all examples presented in this paper, we observe numerically that these sequences are in fact monotone. For small enough values of n , we investigate the qualitative behavior of $M(h)$ for different values of α . We may choose α such that the sequence does not appear to be monotone any longer. A probable advantage is that the computed estimation may be better than the ones obtained with an arbitrary value of α . On the other hand, it will be harder, if not impossible, to extrapolate the data.

4.2.3. Weighted average 2 (WA2). Choose $\alpha \in (0, 1)$ and set

$$h_{ij} := \log \left(\frac{1}{\alpha \text{diam}(F_i \cup F_j) + (1 - \alpha) \max(\delta, \text{dist}(F_i, F_j))} \right).$$

The remarks above apply here too.

4.2.4. Energy minimization (EM). Let ν be the equilibrium measure for E . By definition, $I(\nu) \leq I(\mu)$ for all $\mu \in \mathcal{P}(E)$. A reasonable approach to estimate $I(\nu)$ is to try to minimize $I(\mu)$ over a subset of $\mathcal{P}(E)$. In practice, E may have a complex geometry that makes it difficult to compute the energy of every measure in $\mathcal{P}(E)$. Therefore we instead minimize over a subset of $\mathcal{P}(F)$, where $F := \bigcup_i F_i$, and hope that we still get good estimates for $I(\nu)$. For this to have a chance of working, we should try to ensure that all measures under consideration have their support as close as possible to E , i.e. that $F \setminus E$ be as small as possible. The situation is not as bad as it may look: minimizing over a proper subset of $\mathcal{P}(F)$ gives us a value that is greater than $\log 1/c(F)$, but at the same time, the target value $\log 1/c(E)$ is also greater than $\log 1/c(F)$, and so we can reasonably expect that the errors partially cancel.

For each i , let μ_i be a Borel probability measure on F_i , and assume that any two of them are mutually singular. We consider $\mathcal{Q} := \{ \sum_i t_i \mu_i \mid t \in \Delta_n \} \subset \mathcal{P}(F)$.

Then, for each $\mu \in \mathcal{Q}$, we have

$$\begin{aligned} I(\mu) &= \iint \log \left| \frac{1}{z-w} \right| d\mu(z) d\mu(w) \\ &= \sum_{i,j} t_i t_j \iint \log \left| \frac{1}{z-w} \right| d\mu_i(z) d\mu_j(w) = \sum_{i,j} t_i t_j \tilde{h}_{ij}, \end{aligned}$$

where

$$\tilde{h}_{ij} := \iint \log \left| \frac{1}{z-w} \right| d\mu_i(z) d\mu_j(w).$$

The problem of minimizing $I(\mu)$ over \mathcal{Q} is the quadratic program $Q(\tilde{h})$. As mentioned in §4.1.4, $Q(\tilde{h}) \leq M(\tilde{h})$.

Since $\max(\text{diam}(F_i \cup F_j), \delta) = \text{diam}(F_i \cup F_j)$, we have

$$\log \frac{1}{\text{diam}(F_i \cup F_j)} \leq \min(\tilde{h}_{ij}, \delta) \leq \log \frac{1}{\max(\text{dist}(F_i, F_j), \delta)}.$$

Defining $h_{ij} := \min(\tilde{h}_{ij}, \delta)$, this shows that $a \leq h \leq b$. Thus h is a candidate matrix for approximating $c(E)$.

In practice, we choose the μ_i so that the double integrals defining \tilde{h}_{ij} can be computed explicitly. For example, if the F_i are intervals or squares, we can choose for μ_i the normalized Lebesgue measure on F_i . Another possibility is to choose for μ_i the Dirac measure at a point $x_i \in F_i$. This gives back the midpoint technique described in §4.2.1 above.

Remarks. (i) If $h^{-1}1 \geq 0$ and if h is positive-definite, then the linear and quadratic programs for h have the same solution. Indeed, the Lagrangian

$$L(\lambda, t) := \sum_{i,j} t_i t_j h_{ij} - \lambda \left(\sum_j t_j - 1 \right)$$

has a critical point when $2ht = \lambda$ and $\sum_j t_j = 1$. Since $h^{-1}1 \geq 0$, it follows that $(\lambda, t) := (2, h^{-1}1)/S$ is a solution, as long as we choose S so that $\sum_j t_j = 1$. As h is positive-definite, the minimum is global, and by Proposition 4.1 it is the same as $M(h)$.

(ii) If $F = E$, then

$$I(\nu) \leq \min_{\mu \in \mathcal{Q}} I(\mu) = Q(\tilde{h}) \leq M(\tilde{h}),$$

and hence $M(\tilde{h})$ yields a lower bound for $c(E)$. If $h = \tilde{h}$, then we know that this lower bound converges to $c(E)$. In practice, we may not have $h = \tilde{h}$. When $h \neq \tilde{h}$, both $M(h)$ and $M(\tilde{h})$ give us information: $e^{-M(\tilde{h})}$ is a lower bound for the capacity (possibly better than $e^{-M(a)}$), while $e^{-M(h)}$ converges to it.

4.3. Hardware. All the numerical computations presented in this paper were performed on Pentium 4 and Athlon 64 computers, with clock speed ranging from 2.4GHz to 3.0GHz, and memory ranging from 1GB to 4GB.

5. EXAMPLES

5.1. Union of two intervals. We begin with an example where we know the exact answer. Let $E := [-5, -3] \cup [3, 5]$. Then, using [13, Corollary 5.2.6], we have

$$c(E) = \sqrt{5^2 - 3^2}/2 = 2.$$

TABLE 1. Lower and upper bounds for $c([-5, -3] \cup [3, 5])$.

n	LB	UB	LB time	UB time
2^8	1.91530527	2.08937291	0.6s	0.4s
2^9	1.94905379	2.05293596	2.5s	1.4s
2^{10}	1.96987139	2.03098371	17.8s	5.8s
2^{11}	1.98242280	2.01794238	138.5s	24.2s
2^{12}	1.98986028	2.01029190	1162.0s	104.8s
2^{13}	1.99420728	2.00585379	11034.2s	480.5s
Extrapol	2.00032166	1.99972228		
Rate	0.775	0.786		

Let us see what our numerical methods give.

Let n be an even positive integer, and cover E by n closed intervals of equal length $4/n$. We number them F_1, \dots, F_n from, say, left to right. The quantities $\text{diam}(F_i \cup F_j)$ and $\text{dist}(F_i, F_j)$ are easy to compute. Moreover, since $E \cap F_i$ is an interval of length $4/n$, we can choose $\delta = c(E \cap F_i) = 1/n$. We have everything needed to construct the matrices a and b . We note that both of them are symmetric with respect to their anti-diagonals, and therefore we may as well solve the reduced-size problem, as explained in §4.1.5.

Table 1 shows the lower bounds (LB) and the upper bounds (UB) obtained for different values of n . For the upper bounds, we do not have $a^{-1}1 \geq 0$. However, scaling down E by a factor of 10 gives us $a^{-1}1 \geq 0$, and, for that column, we can compute $M(a)$ by solving a linear system. For the lower bounds, we have to use the linear program approach. This explains the much better times in the UB column.

All calculations for this example have been done with Maple 10 at hardware precision (`Digits:=15`). The commands `LinearSolve` and `LPSolve` are used for computing $M(a)$ and $M(b)$ respectively. We also make use of the new `Compiler:-Compile` command in Maple 10, which enables simple numerical Maple procedures to be compiled in C. This makes the calculations of the entries of a and b much faster. According to Maple documentation, `LinearSolve` and `LPSolve` use hardware-float optimized procedures from the Numerical Algorithms Group (NAG).

We observe that LB and UB are monotone sequences. This suggests extrapolating the data. More precisely, we find numbers A , B and C so that $An^{-B} + C$ best fits the data. The row ‘Extrapol’ in Table 1 shows the value of C , while the row ‘Rate’ shows the value of B . In both cases the convergence rate is faster than the predicted value $1/3$ in the remarks following Theorem 3.2.

We now experiment with intermediate methods described in §4.2. The results are displayed in Table 2. The columns LB and UB are just carried over from Table 1. The computation times for EM, MP and WA2 are similar to those for UB because a linear system was used. For WA1 we have to solve the linear program, and the computation times are similar to those for LB. For WA1 and WA2 we have taken $\alpha = 0.5$ (see §4.2.2 and §4.2.3), and for EM we use normalized Lebesgue measures on the intervals F_i .

We observe that WA1 gives the best approximation for the higher values of n , but at the same time it is the slowest intermediate method. The much faster EM gives

TABLE 2. Intermediate methods for $c([-5, -3] \cup [3, 5])$.

n	LB	EM	WA1	MP	WA2	UB
2^8	1.91530527	1.99983253	2.00129137	2.00282959	2.01390944	2.08937291
2^9	1.94905379	1.99997799	2.00049986	2.00166183	2.00758629	2.05293596
2^{10}	1.96987139	2.00002002	2.00018222	2.00095463	2.00410843	2.03098371
2^{11}	1.98242280	2.00002557	2.00005764	2.00053922	2.00221165	2.01794238
2^{12}	1.98986028	2.00002057	2.00001218	2.00030058	2.00118447	2.01029190
2^{13}	1.99420728	2.00001418	1.99999778	2.00016577	2.00063153	2.00585379
Extr.	2.00032166	–	1.99999111	1.99999078	1.99998684	1.99972228
Rate	0.775	–	1.659	0.824	0.894	0.786
LSR	0.793	0.425*	2.350	0.842	0.901	0.801

the second best approximation. Unfortunately, it appears that the EM sequence is not monotone, so we cannot extrapolate its values. WA1 is monotone for the values of n displayed in the table, but the last entry is less than 2. One can reasonably ask if this entry is correct. It can be recalculated by an independent C++ program using the GLPK package. The difference between the two values turns out to be 4×10^{-8} , confirming that the sequence does indeed drop below 2. Therefore the exponential extrapolation cannot give a better estimate than what we have found for $n = 2^{13}$. Nevertheless, we have included it in the table because, in practice, we do not usually know the exact value of $c(E)$, and so we would have calculated it anyway. The best overall approximation in the table is given by the $n = 2^{13}$ entry of the WA1 sequence. The error is 2.2×10^{-6} . The extrapolations for WA1, WA2 and MP and the last entry of EM are all second-best approximations, as they all lie within 1.4×10^{-5} of the true value.

Knowing that the capacity equals 2 allows us to estimate the convergence rate of a sequence. To do this, we find a least-square fit to the curve $|x(n) - 2| = An^{-B}$ using the last four values of n . The exponent B is shown in the least-square rate (LSR) row (for EM we have used only the last 3 values). WA1 is the fastest converging sequence, but again the slowest to compute.

As discussed in §4.2.4, we could use \tilde{h} instead of h in the EM intermediate method. Let us call this new sequence EM*. We do not know if EM* converges to $c(E)$, but on the other hand we get a monotonically increasing sequence that furnishes lower bounds for $c(E)$. Indeed, here we have $\bigcup_i F_i = E$, and since we refine our partition of E at each step, the solution t found for $n = 2^k$ gives a valid candidate for $n = 2^{k+1}$, namely

$$\left(\frac{t_1}{2}, \frac{t_1}{2}, \frac{t_2}{2}, \frac{t_2}{2}, \dots, \frac{t_{2^k}}{2}, \frac{t_{2^k}}{2} \right) \in \Delta_{2^{k+1}},$$

which has the same associated energy. Therefore, $\min_{\mu} I(\mu)$ can only decrease when k increases. Also, since we observe numerically that \tilde{h} is positive-definite and $\tilde{h}^{-1} \geq 0$, we know that $I(\mu) = M(\tilde{h})$. Another possibility is to directly solve the quadratic program. The results are shown in Table 3.

The EM* sequence gives us much better lower bounds than LB. The entry in row $n = 2^{13}$ is the third best as compared to other intermediate methods for the

TABLE 3. Energy minimization methods for $c([-5, -3] \cup [3, 5])$.

n	LB	EM*	EM	Time
2^8	1.91530527	1.9982042217	1.99983253	0.4s
2^9	1.94905379	1.9991016311	1.99997799	1.4s
2^{10}	1.96987139	1.9995506963	2.00002002	5.9s
2^{11}	1.98242280	1.9997753185	2.00002557	24.0s
2^{12}	1.98986028	1.9998876518	2.00002057	104.4s
2^{13}	1.99420728	1.9999438241	2.00001418	474.8s
Extrapol	2.00032166	2.0000000074	–	
Rate	0.775	0.99986	–	
LSR	0.793	0.99989	0.425	

same n . It falls behind EM, which is not surprising, and WA1. The strength of EM* is that its convergence is very regular. In fact, it looks like the error is proportional to $1/n$. The exponential extrapolation gives $c(E)$ with an error of 7.4×10^{-9} . This is an order of magnitude better than the best approximation found in Table 2. To compute EM*, it is sufficient to solve a linear system. That explains why the times are similar to the times for UB. Note that, in this example, the times for solving the quadratic program are slower than those for the corresponding linear program. For instance, it takes 336 seconds to solve the case $n = 2^{12}$ with `QPSolve`, while it takes 225 seconds for the same case with `LPSolve`. Of course, the three methods (linear system, linear program and quadratic program) return the same answer.

5.2. Union of disk and half-disks. For our second example, we test our method on a compact set E whose capacity is unknown, but was estimated in [14, §7.3]. The set E is made up of a disk and two half-disks:

$$E := \{z : |z| \leq 1\} \cup \{z : |z - 3| \leq 1, \Im(z) \geq 0\} \cup \{z : |z - 3i| \leq 1, \Re(z) \geq 0\}.$$

It was shown in [14] that $c(E) \in [2.18655, 2.21960] =: I^*$. Moreover, the best computed approximation was 2.196961. We will denote this number by c^* .

As suggested at the end of §3.2, we compute $c(\partial E)$ instead of $c(E)$. We also make use of the symmetry of ∂E with respect to $y = x$ to reduce the problem size.

Let n be a positive multiple of 16, and cover ∂E by 4 intervals of length $16/n$ and 12 arcs of length $16\pi/3n$. We number the F_j counterclockwise on each component of ∂E , starting from $2i$, and then $\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$, and finally 2. In this example, computing $\text{diam}(F_i \cup F_j)$ and $\text{dist}(F_i, F_j)$ is not as easy as before. This arises from the fact that an interior point of an arc may be an extremal point. We obtain an efficient implementation by presolving the different cases and then using smart branching. Since $c(F_i)$ is $4/n$ in the case of an interval and $\sin(4\pi/3n)$ in the case of an arc (see [13, Exercice 5.2.4]), we can choose $\delta := \min(4/n, \sin(4\pi/3n)) = 4/n$.

Here, both LB and UB require the use of a linear program. The results for n up to 2^{13} are shown in Table 4. Applying the least-square method of §4.1.4, we can improve the lower bound via a linear system. Using $n = 2^{14}$ and $\lambda = 2$, we find that $c(E) \geq 2.1934415$. In the case of the upper bound, the quadratic program for $n = 2^{14}$ gives $c(E) \leq 2.2003506$. It is interesting to note that, although the interval

TABLE 4. Lower and upper bounds for Example 5.2.

n	LB	UB	LB Time	UB Time
2^{10}	2.1601927	2.2337991	14.2	10.6
2^{11}	2.1762966	2.2174427	89.4	85.3
2^{12}	2.1855031	2.2082671	889.8	689.2
2^{13}	2.1906796	2.2031650	8884.6	5623.6
* 2^{14}	2.1934415	2.2003506	23094.6	15816.4

TABLE 5. Intermediate methods for Example 5.2.

n	WA1	EM*	MP	WA2	EM* Extr.
2^{10}	2.19630416	2.19695120	2.19818413	2.20053834	2.19699402851
2^{11}	2.19662042	2.19697681	2.19760655	2.19878863	2.19699379405
2^{12}	2.19679610	2.19698700	2.19730715	2.19789958	2.19699373479
2^{13}	2.19689041	2.19699105	2.19715323	2.19744984	2.19699372105
2^{14}	2.19694045	2.19699266	2.19707458	2.19722301	2.19699371801
2^{15}	—	2.19699330	2.19703467	2.19710907	2.19699371734
Extr.	2.19699699	2.19699372	2.19699321	2.19699315	2.19699371717
Rate	0.914	1.333	0.976	0.990	2.210

$[2.1934415, 2.2003506]$ is 5 times smaller than I^* , the approximation c^* is still valid and almost right in the middle of it. We deduce that the error on c^* is less than 3.5×10^{-3} , which is better than the 2.3×10^{-2} computed in [14].

To obtain approximations to $c(E)$, we experiment with the techniques MP, WA1, WA2 and EM*. The first three are easy to implement. MP and WA2 can be calculated with a linear system (though we have to consider $\frac{1}{3}E$), while WA1 requires a linear program. EM* is, not surprisingly, the hardest to implement. The double integrals cannot be computed explicitly (except in the case of parallel intervals), and we have to resort to numerical integration. For the case of two different F_i , we use an adaptive multidimensional integration Fortran routine, the so-called ACM Algorithm 698, as described in [2]. The case of integration over $F_i \times F_i$ is solved by first reducing symbolically the double integral to a simple integral, and then employing a Taylor series. For EM*, a linear system is sufficient. The results for the intermediate methods are shown in Table 5.

WA1 and WA2 are computed using $\alpha = 0.5$. We are unable to solve the linear program (WA1) for $n = 2^{15}$ because we run out of memory. As noted in the previous example, the sequence EM* furnishes lower bounds for $c(E)$. Moreover, these lower bounds are faster to compute than the corresponding values of LB, they require less processing memory and they are much better. For example, for $n = 2^{13}$ it takes 8885 seconds to compute LB, while it takes only 1526 seconds to compute EM*. Even though the matrix for EM* takes longer to build, the fact that a linear system suffices for this sequence is predominant.

On top of that, EM* is the fastest converging sequence. It is also very regular, in the sense that the sequence of extrapolations for EM* (last column of Table 5) looks like a monotone decreasing sequence. This suggests fitting an exponential curve to that sequence.

Conclusion. We have proved that $c(E) \in [2.1969933, 2.2003506]$, and our best guess is $c(E) \approx 2.19699371717$. (The gap between this value and c^* is less than 3.3×10^{-5} .)

5.3. Cantor set. Let $E_0 := [0, 1]$. For $k \geq 1$, we recursively define

$$E_k := \frac{1}{3}E_{k-1} \cup \left(\frac{1}{3}E_{k-1} + \frac{2}{3} \right).$$

Then $E_k \subset E_{k-1}$ and we set $E := \bigcap_k E_k$. This is the classical middle-third Cantor set: a perfect, totally disconnected set of Lebesgue measure zero, but of positive capacity. The exact value of $c(E)$ is unknown. Since $E \subset [0, 1/3] \cup [2/3, 1]$, we have

$$c(E) \leq \frac{1}{2} \sqrt{\left(\frac{1}{2}\right)^2 - \left(\frac{1}{6}\right)^2} = \frac{1}{3\sqrt{2}} \approx 0.23570226.$$

At the other end, it can be shown that $c(E) \geq 1/9$ (see e.g. [13, Theorem 5.3.7]). In this paragraph, we shall significantly reduce the gap between these bounds and obtain a good approximation to $c(E)$.

Let $n := 2^k$ for some positive integer k . We choose for F_i the intervals of E_k labelled from left to right. We make use of the symmetry of E about $1/2$ and reduce the problem size accordingly. The functions *dist* and *diam* are straightforward to compute, at least in theory. For large values of k , we must take care of the cancellation phenomenon that appears when computing the distance between two nearby points of E . We get around the problem by using integers instead of floats for this part of the calculation. As each set $F_i \cap E$ is a scaled version of E and $c(E) \geq 1/9$, we can take $\delta := 3^{-k}/9$.

Here, both LB and UB can be computed via a linear system. In order to compute $M(h)$ for large values of k , we implement a matrix-free iterative method, namely the conjugate gradient. This allows a trade-off between memory and computation time. To accelerate the convergence, we use a diagonal preconditioner of typical block size between 2^{10} and 2^{12} . These blocks are LU-factored.

The four highest values of n computed are shown in Table 6. A better option for δ is to take $\delta = 3^{-k}c_0$, where c_0 is any precomputed lower bound for $c(E)$. If we apply this and recompute LB for $n = 2^{19}$ using the lower bound just obtained for this value of n , we get an improved lower bound as shown in the last row. This entry is a little bit faster to compute, since we can start our iterative method from the previous solution vector t .

To get an approximation to $c(E)$, we compute MP and EM*, both via a linear system. For MP we take $\delta = 3^{-k}0.22094810685$. All integrals involved in EM* are symbolically precomputed. The resulting expression for the case $F_i \times F_i$ is prone to cancellation, and we have to take special care to compute it to the desired precision. The results are shown in Table 7.

In this example, EM* does not, *a priori*, give lower bounds for $c(E)$, because $F_i \not\subset E$. Nevertheless, it looks like the sequence is still increasing. At the same time, we observe that MP is a decreasing sequence. In the light of these observations, we can reasonably conjecture that both of them furnish bounds for $c(E)$. We have no proof of this.

TABLE 6. Lower and upper bounds for the classical Cantor set.

n	LB	UB	Time
2^{16}	0.22093838586	0.22096249759	2h 00m
2^{17}	0.22094362271	0.22095595104	8h 58m
2^{18}	0.22094630066	0.22095260369	1d 11h 45m
2^{19}	0.22094766991	0.22095089228	7d 6h 40m
$* 2^{19}$	0.22094810685		5d 17h 46m

TABLE 7. Intermediate methods for the classical Cantor set.

n	EM*	MP	EM* Extr.	MP Extr.
2^{15}	.220949078827933	.220949238296649	.220949102189205	.220949102189264
2^{16}	.220949090247155	.220949171767077	.220949102189468	.220949102189483
2^{17}	.220949096084625	.220949137757366	.220949102189531	.220949102189550
2^{18}	.220949099068712	.220949120371698	.220949102189488	.220949102189510
2^{19}	.220949100594174	.220949111484188	.220949102189537	.220949102189488
Ex.	.220949102189537	.220949102189488	.220949102189506	.220949102189508
Ra.	0.968	0.968	(average 2^{16} – 2^{19})	(average 2^{16} – 2^{19})

The extrapolation sequences for EM* and MP are monotonically increasing up to $n = 2^{17}$ (previous values of n are omitted in the table). Then they start to oscillate. It is probable that this is due to cumulative rounding errors. The matrix size is very large for these values of n , and double precision cannot guarantee more than 15 correct digits. Noting that the numbers in Table 7 all have 15 digits, we conclude that we have reached the hardware precision in this example. It makes little sense to extrapolate the last two columns. Instead we have simply calculated the average of the last four entries.

Conclusion. We have proved that $c(E) \in [0.22094810685, 0.22095089228]$, and our best guess is $c(E) \approx 0.220949102189507$.

Remark. It was shown in [17] that the Green’s function for E_k , and therefore its capacity, can be derived from a Schwarz-Christoffel conformal map. For $k \geq 2$, the conformal map cannot be determined analytically, and numerical integration is necessary. Driscoll’s MATLAB Schwarz-Christoffel Toolbox [6] is an efficient package for computing Schwarz-Christoffel conformal mappings. Embree and Trefethen have put these ideas together in [7, Section 4], where they discuss the problem of computing the capacity. Using their method, Embree, Trefethen and Banjai have computed $c(E_k)$ for k from 1 to 9 (private communication). Their guess from extrapolation is $c(E) \approx 0.2209491$, which agrees with our proposed value.

5.4. Cantor square. As mentioned at the end of §3.2, the smaller the Minkowski dimension of E , the smaller the number of sets F_k of a given diameter needed to cover E , thereby reducing the problem size. In Examples 5.1 and 5.2 we had $\dim(E) = 1$, while in Example 5.3 we had $\dim(E) = \log 2 / \log 3 < 1$. We now try our method on an example where $\dim(E) > 1$.

TABLE 8. Lower and upper bounds for the Cantor square.

n	LB	UB	Time
2^{10}	0.562159	0.585763	0s
2^{12}	0.569368	0.579012	7s
2^{14}	0.572344	0.576226	297s
2^{16}	0.573550	0.575095	51 261s

Let $E_0 := [0, 1] \times [0, 1]$. For $k \geq 1$, we recursively define

$$E_k := \frac{1}{3}E_{k-1} + \left\{0, \frac{2}{3}, \frac{2}{3}i, \frac{2}{3} + \frac{2}{3}i\right\}.$$

Then $E_k \subset E_{k-1}$, and we set $E := \bigcap_k E_k$. The compact set E is the cartesian product of the classical middle-third Cantor set with itself. It is well known that $\dim(E) = \log 4 / \log 3$.

Since E is a subset of the unit square, we have (see [13, p.135])

$$c(E) \leq c([0, 1]^2) = \frac{\Gamma(1/4)^2}{4\pi^{3/2}} \approx 0.59017.$$

Also, since $c(K \times K) \geq 2c(K)$ for any compact subset K of \mathbb{R} (see e.g. [13, p.169]),

$$c(E) \geq 2 \times 0.22094810685 \approx 0.4418962137.$$

Let $n := (2^k)^2$ for an integer $k \geq 2$. We choose for F_i the squares of E_k . Our compact set in this example exhibits more symmetries than in the previous examples. Calculations show that it is sufficient to consider just the F_i that intersect the triangle $\{z : 0 \leq \Re(z) \leq \frac{1}{2}, 0 \leq \Im(z) \leq \Re(z)\}$. There are $m := 2^{k-2}(2^{k-1} + 1)$ such F_i . Thus the problem size can be reduced by a factor close to 8.

The functions dist and diam are easy to compute. As each set $F_i \cap E$ is a scaled version of E , we can take $\delta_k := 3^{-k}c_0$, where c_0 is a previously calculated lower bound for $c(E)$. When computing the LB sequence, we start with $\delta_2 := 3^{-2}0.3124678093$, and update the value of δ between each k . Both LB and UB require a linear program. Table 8 displays the results.

We can compute MP and EM* via a linear system (see Table 9). All integrals involved in EM* are symbolically precomputed. The ‘Time’ column is for the EM* sequence; the times for MP turn out to be approximately twice as fast. For both intermediate methods we have extrapolated the extrapolation sequence.

Conclusion. We have proved that $c(E) \in [0.573550, 0.575095]$, and our best guess is $c(E) \approx 0.5743450704$.

5.5. Comparison of the examples. We conclude with some remarks concerning the fact that the convergence is slower in this last example, at least in regard to the required computational effort. For $n = 2^{16}$, our confidence interval around $c(E)$ has relative length 0.27%. Since, for this value of n , the matrix dimensions are 8256×8256 (after reduction), the closest corresponding value for the linear Cantor set is $n = 2^{14}$. There, the confidence interval has relative length 0.042%. Hence, for the same problem size, the precision is 6.4 times better in the linear Cantor set case. Moreover, this does not take into account the slower times for the Cantor square case due mainly to the fact that we have to solve a linear program.

TABLE 9. Intermediate methods for the Cantor square.

n	EM*	MP	EM* Extr.	MP Extr.	Time
2^8	0.5701134095	0.5659617752	0.5744538404	0.5744899295	0s
2^{10}	0.5727536391	0.5711913783	0.5743546783	0.5743623960	0s
2^{12}	0.5737469433	0.5731595871	0.5743460312	0.5743473754	4s
2^{14}	0.5741203109	0.5738995670	0.5743451770	0.5743453879	67s
2^{16}	0.5742606163	0.5741776675	0.5743450822	0.5743451137	1135s
Extr.	0.5743450822	0.5743451137	0.5743450704	0.5743450697	
Rate	0.706	0.706	1.586	1.429	

TABLE 10. Comparison of convergence rates.

	Ex. 5.1	Ex. 5.2	Ex. 5.3	Ex. 5.4
UB Rate w.r.t. n	0.786	0.847	0.968	0.650
UB Rate w.r.t. d	0.786	0.847	0.611	0.820

In Table 10 we compare the UB convergence rate for all our examples. We choose UB instead of LB because *a priori* estimations of the capacity are not involved. The first row shows that the convergence rate with respect to the problem size n (the number of sets F_i) is significantly slower for Example 5.4 than for the other examples. In the second row we have computed the convergence rate with respect to $d := \max_i \text{diam}(F_i)$. This is obtained from the first row by multiplying by $\dim(E)$. We observe that, this time around, the linear Cantor set exhibits the slowest convergence. This suggests that its α exponent (see §3.2) might be significantly smaller than $1/2$, which is the exponent for Examples 5.1 and 5.2.

6. GENERALIZED CANTOR SETS

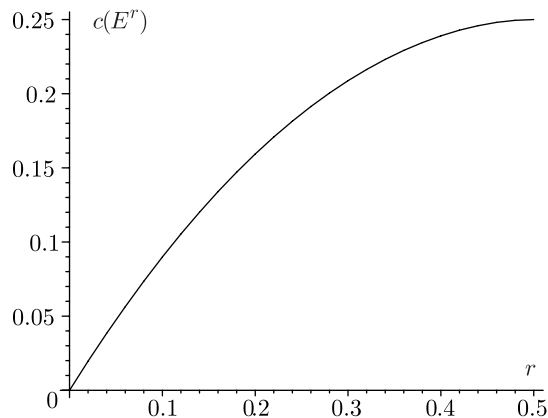
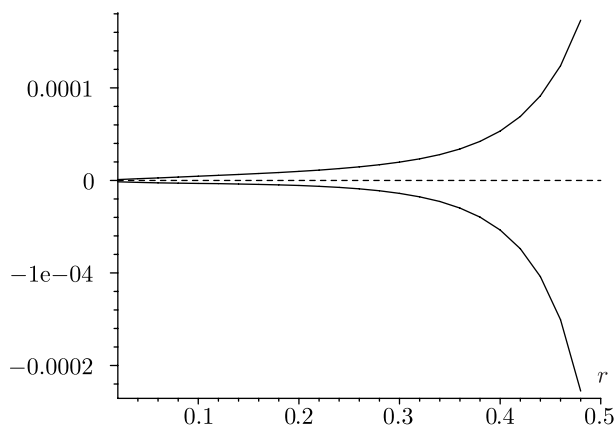
In this final section, we experiment on a generalization of the linear Cantor set. Fix a real number $r \in (0, 1/2)$. Let $E_0^r := [0, 1]$. For $k \geq 1$, we recursively define

$$E_k^r := rE_{k-1}^r \cup (rE_{k-1}^r + 1 - r).$$

Then $E_k^r \subset E_{k-1}^r$ and we set $E^r := \bigcap_k E_k^r$. The choice $r = 1/3$ returns the classical Cantor set. We are interested in the behavior of $c(E^r)$ as a function of r .

The implementation of our algorithm is very similar to that in §5.3. In fact, the only difference is in the routine that computes the distance between two endpoints of intervals of E_k^r . As mentioned before, we must take some precautions to avoid cancellation. Since we can no longer use integers for general values of r , other strategies have to be employed. The smallest distances are precomputed and stored in an array whose typical size is 2^{10} . Then, using a binary representation of the intervals of E_k^r , we can compute the required distance by using the table at most twice (if $n \leq 2^{20}$).

For $n = 2^{15}$ we have computed the quantities LB, UB and EM* for each r in $\{\frac{1}{50}, \frac{2}{50}, \dots, \frac{24}{50}\}$. Figure 1 shows the graph of $c(E^r)$. Actually, this is the graph of EM*, where linear segments are used between the data points and where we have added the points $(0, 0)$ and $(1/2, 1/4)$.

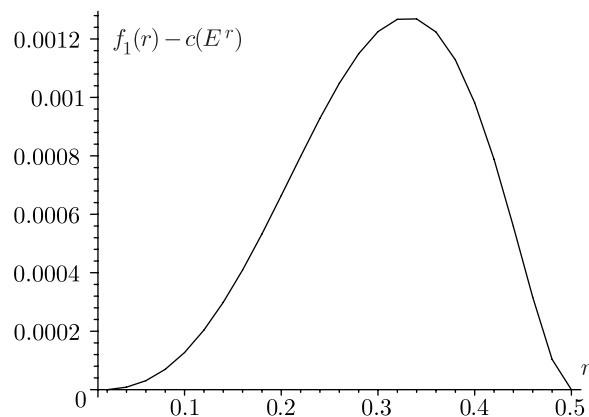
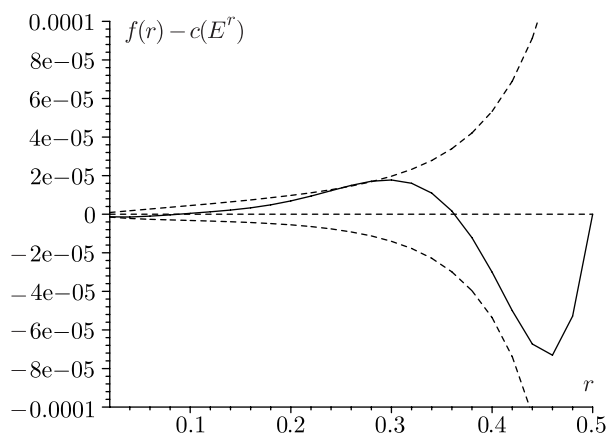
FIGURE 1. Graph of $c(E^r)$.FIGURE 2. Upper bounds and lower bounds on the error on $c(E^r)$.

The graphs of LB and UB are so close to $c(E^r)$ that they would be indistinguishable if added to Figure 1. Instead, we have plotted the differences LB-EM* and UB-EM* in Figure 2.

The curve shown in Figure 1 looks like a parabola. We observe that the slope at the origin is close to 1 and the slope at $1/2$ is close to 0. This suggests comparing with the curve with $f_1(r) := r(1-r)$. The difference $f_1(r) - c(E^r)$ is shown in Figure 3.

The first thing to note is that $f_1(r)$ is not a bad approximation to $c(E^r)$. The maximal error is less than 1.3×10^{-3} . We remark that this maximum is attained at a value of r near $1/3$. Since $1/3$ is a point of symmetry in our problem, its appearance here may not be fortuitous. Moreover, the function $f_1(r) - c(E^r)$ is non-negative throughout $[0, 1/2]$ and has a simple shape. These facts, taken together, suggest going one step further and guessing a simple expression for $f_1(r) - c(E^r)$.

We look for a function f_2 of the form $f_2(r) := kr^\alpha(1/2 - r)^\beta$. Imposing the constraint that f_2 has a local maximum at $r = 1/3$ gives $\alpha = 2\beta$. It then suffices to choose two points on the estimated curve for $c(E^r)$ to get estimates for k and β .

FIGURE 3. The graph of $f_1(r) - c(E^r)$.FIGURE 4. The graph of $f(r) - c(E^r)$.

For example, with $r = 8/50$ and $r = 17/50$ we get $k \approx 0.5058$ and $\beta \approx 1.5007$. We therefore propose the following simple expression as an approximation to $c(E^r)$:

$$f(r) := r(1-r) - \frac{r^3}{2} \left(\frac{1}{2} - r \right)^{3/2}.$$

Figure 4 displays the graph of $f(r) - c(E^r)$, together with the error bounds on $c(E^r)$. An estimation of the capacity of the middle-third Cantor set is given by

$$f(1/3) = \frac{432 - \sqrt{6}}{1944} \approx 0.220962197.$$

According to §5.3, this value is correct up to 1.3×10^{-5} .

The function $f(r)$ gives a surprisingly good approximation to $c(E^r)$, considering its very simple form. Is it the beginning of a series development for $c(E^r)$?

REFERENCES

1. L. Baribeau, D. Brunet, T. Ransford, J. Rostand, *Iterated function systems, capacity and Green's functions*, Comput. Methods Funct. Theory **4** (2004), 47–58. MR2081665 (2005f:31003)
2. J. Berntsen, T. O. Espelid, A. Genz, *Algorithm 698: DCUHRE: an adaptive multidimensional integration routine for a vector of integrals*, ACM Trans. Math. Software **17** (1991), 452–456. MR1140035
3. D. G. Cantor, *On an extension of the definition of transfinite diameter and some applications*, J. Reine Angew. Math. **316** (1980), 160–207. MR581330 (81m:12002)
4. L. Carleson, T. W. Gamelin, *Complex dynamics*, Springer, New York, 1993. MR1230383 (94h:30033)
5. L. Carleson, V. Totik, *Hölder continuity of Green's functions*, Acta Sci. Math. (Szeged) **70** (2004), 557–608. MR2107529 (2005h:30052)
6. T. A. Driscoll, *Algorithm 756: A MATLAB toolbox for Schwarz-Christoffel mapping*, ACM Trans. Math. Software, **22** (1996), 168–186; <http://www.math.udel.edu/~driscoll/SC>.
7. M. Embree, L. N. Trefethen, *Green's functions for multiply connected domains via conformal mapping*, SIAM Rev. **41** (1999), 745–761. MR1722999 (2000h:30053)
8. K. J. Falconer, *Fractal geometry. Mathematical foundations and applications*, 2nd ed., Wiley, Chichester, 2003. MR2118797 (2006b:28001)
9. S. Mehrotra, *On the Implementation of a Primal-Dual Interior Point Method* SIAM J. Optimization **2** (1992), 575–601. MR1186163 (93g:90047)
10. Ch. Pommerenke, *Univalent functions*, Vandenhoeck & Ruprecht, Göttingen, 1975. MR0507768 (58:22526)
11. Ch. Pommerenke, *Uniformly perfect sets and the Poincaré metric*, Arch. Math. **32** (1979), 192–199. MR534933 (80j:30073)
12. Ch. Pommerenke, *Boundary behaviour of conformal maps*, Springer, Berlin, 1992. MR1217706 (95b:30008)
13. T. Ransford, *Potential theory in the complex plane*, Cambridge University Press, Cambridge, 1995. MR1334766 (96e:31001)
14. J. Rostand, *Computing logarithmic capacity with linear programming*, Experiment. Math. **6** (1997), 221–238. MR1481591 (98f:65032)
15. J. L. Walsh, *Interpolation and approximation by rational functions in the complex domain*, American Mathematical Society, Providence RI, 1969. MR0218588 (36:1672b)
16. J. Wang, *The theory of games*, Oxford University Press, Oxford, 1988. MR0969605 (90a:90219)
17. H. Widom, *Extremal polynomials associated with a system of curves in the complex plane*, Adv. in Math. **3** (1969), 127–232. MR0239059 (39:418)

DÉPARTEMENT DE MATHÉMATIQUES ET DE STATISTIQUE, UNIVERSITÉ LAVAL, QUÉBEC (QC), CANADA G1K 7P4

E-mail address: ransford@mat.ulaval.ca

DÉPARTEMENT DE MATHÉMATIQUES ET DE STATISTIQUE, UNIVERSITÉ LAVAL, QUÉBEC (QC), CANADA G1K 7P4

E-mail address: jrostand@mat.ulaval.ca