Maple Transactions

Log-lightning computation of capacity and Green's function

PETER J. BADDOO, Massachusetts Institute of Technology, USA
LLOYD N. TREFETHEN, University of Oxford, UK

Abstract. A basic measure of the size of a set E in the complex plane is the logarithmic capacity $\mathrm{cap}(E)$. Capacities are known analytically for a few simple shapes like ellipses, but in most cases they must be computed numerically. We explore their computation by the new "log-lightning" method based on reciprocal-log approximations in the complex plane. For a sequence of 16 examples involving both connected and disconnected sets E, we compute capacities to 8–15 digits of accuracy at great speed in MATLAB. The convergence is almost-exponential with respect to the number of reciprocal-log poles employed, so it should be possible to compute many more digits if desired in Maple or another extended-precision environment. This is the first systematic exploration of applications of the log-lightning method, which opens up the possibility of solving Laplace problems with an efficiency not achievable by previous methods. The method computes not just the capacity, but also the Green's function and its harmonic conjugate. It also extends to "domains of negative measure" and other Riemann surfaces.

CCS Concepts: • **Mathematics of computing** → **Partial differential equations**.

Additional Key Words and Phrases: Laplace equation, Green's function, capacity, log-lightning solver

Recommended Reference Format:
Peter J. Baddoo and Lloyd N. Trefethen. 2021. Log-lightning computation of capacity and Green's function. *Maple Trans.* 1, 1, Article 5 (July 2021), 13 pages. https://mapletransactions.org/index.php/maple/article/view/14124

1 Introduction

The capacity $\mathrm{cap}(E)$, also known as the logarithmic capacity, is a measure of the size of a nonempty compact set E in the complex plane, which may contain several components but no isolated points. The simplest way to understand this number comes from its physical interpretation. In two-dimensional potential theory, a unit point charge at z_0 is associated with the potential $\log|z - z_0|$. Suppose a quantity Q of charge is injected into E and allowed to find its minimal-energy configuration. This will be an equilibrium distribution (the *equilibrium measure*) in which all of E is at a constant potential. As $z \longrightarrow \infty$, the potential will be indistinguishable from that of a point charge, being asymptotic to $q \log|z|$ for some q, and $\mathrm{cap}(E)$ is Q/q, the amount of charge Q on E corresponding to a point charge of strength $q = 1$.

Mathematically, $\mathrm{cap}(E)$ can be defined in terms of the Green's function of E, the harmonic function G in $C \cup \{\infty\}\backslash E$ with $G(z) \longrightarrow 0$ as $z \longrightarrow \partial E$ and $G(z) \sim \log|z|$ as $z \longrightarrow \infty$. The defining formula is

$$G(z) = \log|z| - \log \mathrm{cap}(E) + o(1), \quad z \longrightarrow \infty, \tag{1}$$

and a rigorous presentation can be found in the outstanding textbook by Ransford [15]; see also the survey [16]. The capacity arises in numerous applications in potential theory, complex analysis,

Authors' addresses: Peter J. Baddoo, baddoo@mit.edu, Massachusetts Institute of Technology, Cambridge, MA, USA, 02139; Lloyd N. Trefethen, trefethen@maths.ox.ac.uk, University of Oxford, Mathematical Institute, Woodstock Road, Oxford, UK, OX2 6GG.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others must be honored. Abstracting with credit is permitted.
© 2021 Copyright held by the owner/author(s). Publication rights licensed to Maple Transactions, under Creative Commons CC-BY 4.0 License.
https://mapletransactions.org/index.php/maple/article/view/14124

and approximation theory. For example, the reason the leading coefficients of the Chebyshev polynomials $T_n(x)$ grow at a rate proportional to 2^n as $n \to \infty$ is that the interval $[-1, 1]$ has capacity $1/2$.

An interval of length L has capacity $L/4$, and a disk of radius r has capacity r. These are special cases of the formula $\mathrm{cap}(E) = (a + b)/2$ for an ellipse of semi-axes a and b. Explicit formulas are also known for a few other special domains E, as listed in Table 5.1 of [15], but in most cases, $\mathrm{cap}(E)$ must be computed numerically. The aim of this paper is to explore the computation of these numbers by the new *log-lightning* method introduced in [12]. If E has an analytic boundary, then exponentially-convergent approximations are possible via degree n polynomial approximations, as described in [19]. Most domains arising in applications have corners, however, where singularities of the Green's function almost invariably appear, and polynomial methods converge too slowly in such cases to provide more than a few digits of accuracy (say, 2 or 3). A few years ago the *lightning Laplace* method was introduced for such problems, based on rational approximations, which improves the convergence rate dramatically to root-exponential ($O(\exp(-C\sqrt{n}))$ with $C > 0$) [2, 9]. The newer log-lightning approach explored here, based on reciprocal-log approximations, has the prospect of further acceleration to what we call *almost-exponential convergence*, meaning at a rate $O(\exp(-Cn/\log n))$ for some $C > 0$. For intermediate accuracies (say, 5 or 6 digits), there may be little gain, but for high accuracies (say, 10 digits or more), the new method appears to offer a decisive advantage. We shall also show that it can treat slits or arcs and even Riemann surfaces and "domains of negative measure" in a natural way.

The determination of capacities is an interesting problem in itself, and at the same time, we view it as a proxy for planar Laplace problems more generally. Our computed examples will highlight that not just the number $\mathrm{cap}(E)$ is computed, but also the Green's function and its harmonic conjugate, which is a multivalued function. The log-lightning technique can certainly be extended to the Stokes equations of viscous fluid flow, since these can be reduced to the Laplace equation [4]. We do not yet know if it can be extended to the Helmholtz equation of electromagnetic and acoustic wave propagation [10].

2 The log-lightning method

As described in [12], the starting point of reciprocal-log approximation is the fact that degree n minimax rational approximations to e^{as} ($a > 0$) for $s \in (-\infty, 0]$ converge exponentially as $n \to \infty$ [5]. (Minimax means optimal in the supremum norm, and degree n means at most n poles, counted with multiplicity and including poles at ∞.) By the change of variables $z = e^s$, $s = \log z$, it follows that the function z^a can be approximated on $[0, 1]$ with exponential convergence by rational functions of $\log z$, despite having a branch point singularity at $z = 0$ if a is not an integer. This ordinary enough fact has extraordinary implications for approximation of analytic functions with branch point singularities, typically at corners, on domains in the complex plane. To solve Laplace problems, we will approximate boundary data by the real parts of these analytic functions.

The log-lightning method does not attempt to compute minimax approximations, but works with simpler approximations with comparable convergence rates. For z^a on $[0, 1]$, exponential convergence can be achieved by approximants of the form $p((\log z - \sigma n)^{-1})$, where p is a polynomial of degree n and σ is any fixed positive number. Note that these are confluent approximations: all the poles are fixed at the same point $z = e^{\sigma n}$, corresponding to $s = \sigma n$ in the s-plane. In this paper all our computations are in this confluent mode, and in fact, following numerical experience, we always use the value $\sigma = 1/2$. For the alternative possibility of poles spread out along a Hankel contour in the s-plane, which corresponds to poles on many sheets of a Riemann surface in the z-plane, see [12].

To start with the base case, suppose we want to approximate a function f on a connected compact domain E bounded by analytic arcs meeting at corners $z_1, \dots, z_m$. We assume f is analytic on E apart from branch point singularities at $\{z_k\}$, and we assume E has a nonempty interior including (without loss of generality) the point $z = 0$. Following eq. (3.2) of [12], we work with *reciprocal-log approximations* to f of the form

$$g(z) = p_0\Big(\frac{1}{z}\Big) + \sum_{j=1}^{m} p_j\Big(\frac{1}{\log((z - z_j)/z) - \frac{1}{2}n_j}\Big), \tag{2}$$

where each p_j is a polynomial of degree n_j, $0 \le j \le m$. Each term $p_j(\cdots)$ for $j \ge 1$ has branch points at $z = 0$ and $z = z_j$ and an order n_j pole near the origin at $-z_j e^{-n_j/2}/(1 - e^{-n_j/2})$ but is analytic for $z \in C \cup \{\infty\}\backslash E$. (We assume n_j is large enough that the pole lies in the interior of E. In coding, some care may be needed to ensure that $\log((z - z_j)/z)$ takes a single branch outside E.) It is shown in [12] that under very general assumptions, approximations of this form exist with almost-exponential convergence as $n \longrightarrow \infty$, where $n = \min n_j$.

For the application to capacities and Green's functions, what we need is not a complex analytic function but its real part, a harmonic function. Specifically, suppose we find a function u harmonic in $C \cup \{\infty\}\backslash E$ that extends to the boundary ∂E with

$$u(z) = \log|z|, \quad z \in \partial E. \tag{3}$$

Then (1) implies $u(z) = \log|z| - G(z)$ and

$$\operatorname{cap}(E) = e^{u(\infty)}. \tag{4}$$

And now we can define our log-lightning approximation problem: find (complex) coefficients of the polynomials p_j of (2) such that the real part,

$$u(z) = \operatorname{Re} g(z), \tag{5}$$

satisfies (3) to high accuracy. Equation (4) then gives the capacity, and we immediately have the harmonic conjugate of u,

$$v(z) = \operatorname{Im} g(z). \tag{6}$$

Note that the space of approximations (2) is linear, since the locations of the singularities are fixed by the parameters $n_0, \dots, n_m$. We solve (3) numerically by linear least-squares fitting in a set of sample points along ∂E with exponential clustering near the corners. Typically there are hundreds or thousands of these sample points, so the least-squares problem involves matrices of dimensions on the order of a few thousand by a few hundred. This presents no difficulty for standard linear algebra software of the kind invoked for example by the backslash command in MATLAB, where the computation `x = A\b` takes around 0.01 secs. on a laptop if A has dimensions 2000 × 200.

One further algorithmic aspect of the log-lightning method must be mentioned. Mathematically, the columns of the least-squares matrix A could be constructed by sampling monomial bases for the polynomials p_j of (2): powers of z^{-1} for $j = 0$ and powers of $(\log((z - z_j)/z) - \frac{1}{2}n)^{-1}$ for $j \ge 1$. However, this is numerically unstable for familiar reasons of Vandermonde matrices. To get more than a few digits of accuracy in most problems it is crucially important to use the stable alternative known as Vandermonde with Arnoldi orthogonalization introduced in [3], and we do this in all our computations, making use of the codes `VAorthog` and `VAeval` provided in [4].

3 Connected domains with interior

We first demonstrate the log-lightning method on the example of a square with vertices $(\pm 1, \pm 1)$, whose exact capacity is $\operatorname{cap}(E) = \Gamma(1/4)^2/(2\pi^{3/2}) \approx 1.1803$. The following MATLAB code (requiring `VAorthog` and `VAeval`) computes $\operatorname{cap}(E)$ to 15-digit accuracy in 0.01 secs. on a laptop.

Experiments show that for this problem the polynomial term is not needed, so it has not been included. This formulation exploits the fourfold symmetry so as to end up with a 300×52 rather than 1200×208 least-squares problem. (This makes surprisingly little difference, less than a factor of 2 in the overall timings.)

```
M = 300; s = tanh(linspace(-14,14,M)');         % clustered points in [-1,1]   01
Z = 1i-s; Z = [Z; 1i*Z; -Z; -1i*Z];             % sample points on square      02
F = log(abs(Z));                                % RHS for least-squares prob   03
n = 25;                                         % # poles near each corner     04
[H,Q] = VAorthog(1./(log(1-(1+1i)./Z)-n/2),n);  % log-lightning columns        05
cs = @(x,k) circshift(x,k);                     % convenient abbreviation      06
A = Q + cs(Q,M) + cs(Q,2*M) + cs(Q,3*M);        % imposing fourfold symmetry   07
A = A(1:M,:); F = F(1:M);                       % exploiting symmetry further  08
c = reshape([real(A) -imag(A)]\F,[],2)*[1;1i];  % least-squares coeff vector   09
uinf = real(4*VAeval(-2/n,H)*c);                % value of solution at infty   10
capacity = exp(uinf)                            % capacity                     11
exact = gamma(1/4)^2/pi^(3/2)/2                 % exact value                  12
```

Let us comment on each line of the code. Lines 1 and 2 discretize the square by 300 points exponentially clustered near the corners along each side, and line 3 evaluates the right-hand side of (3). Line 4 fixes the number of confluent singularities near each corner at 25, and line 5 applies Vandermonde with Arnoldi orthogonalization to construct a matrix Q of orthogonalized powers near one of the corners. Lines 6 and 7 add up the four rotations of these columns to make the fourfold-symmetric complex fitting matrix A, which has 1200 rows and 26 columns at this stage (it would be 104 columns if we had not exploited symmetry). Line 8 exploits the symmetry further by reducing back to 300 rows since each quartile of 300 rows will be equivalent. Line 9 splits the matrix into real and imaginary parts, since our aim is to fit the data by a real harmonic function, and invokes MATLAB backslash to solve the linear least-squares problem, which now has size 300×52. It then converts the result into the complex coefficient vector c associated with the solution u as well as its harmonic conjugate. Line 10 applies (2) to evaluate $u(\infty) = \mathrm{Re}\, g(\infty)$, line 11 gets the capacity from (4), and line 12 calculates the analytical solution for comparison. On our laptop the code prints the numbers

```
capacity = 1.180340599016083
   exact = 1.180340599016096
```

The elapsed time of 0.01 secs. makes this computation more than 100 times faster than the result reported for the same problem in [11].

Figure 1 gives further information about log-lightning computations on the square. In the plot on the right, we see the difference between the root-exponential convergence of the lightning method and the nearly-exponential convergence of the log-lightning method. This holds down to accuracy 10^{-14} or so, after which the results are jagged. We believe this is due to ill-conditioning of the matrix A caused by lack of orthogonality between the columns associated with one corner and those associated with another. The plot in the middle of the figure is of a kind we draw routinely to evaluate the success of these computations. It shows the errors at all the sample points (for $n = 35$), revealing an even pattern at a level below 10^{-12}.

Figure 2 shows six more examples of Green's functions for connected domains with interior, which we will discuss columnwise. All the computations make use of grids exponentially clustered near the corners according to formulas along the lines of `s = tanh(linspace(-15,15,300)')`.

In the first column, the first image shows a lens with corners ±1 of half-angle $\pi/4$. A computation with $n_0 = 0$ and $n_1 = n_2 = 15$ reproduces the exact result $\mathrm{cap}(E) = 2/3$ with error 5.7×10^{-12}. The second image shows a lune defined by angles $\pi/2$ and $\pi/3$ relative to the chord $[-1, 1]$. A

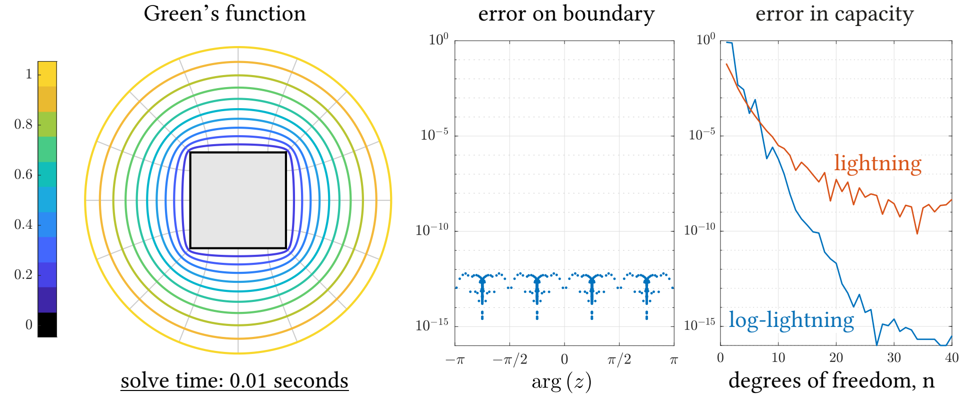

Fig. 1. Log-lightning computation of the Green's function for a square. The method converges exponentially down to an accuracy of about 10^{-12} in the Green's function, measured all the way up to the singular corners (middle plot). With $n = 30$ the scalar $\mathrm{cap}(E)$ is accurate to 10^{-15}. The contour plot also shows level curves of the harmonic conjugate (6) (faint gray).

computation with $n_0 = 0$ and $n_1 = n_2 = 15$ reproduces the exact result $\mathrm{cap}(E) = 3/(5\sin(3\pi/10))$ with error 1.7×10^{-11}.

The second column of Figure 2 turns to the unit square with one (above) and two (below) quarter-circular bites removed. Degrees $n_j = 16$ ($j \ge 0$) suffice to match the correct results of about 1.0883504816708 and 0.9961224476807 with errors around 10^{-13}.

The third column of the figure shows connected domains each consisting of two lobes that just touch at a point. The first is the lemniscate defined by $|z^2 - 1| = 1$, whose capacity is known to be 1. Using $z_c = 0.5$ as the center point for expansions, we get 12-digit accuracy with $n_0 = 0$ and $n_1 = n_2 = 80$. The second is a distortion of the lemniscate in which the left lobe is tilted down 45° and the right lobe is enlarged by a factor of 1.5. This makes the capacity about 0.863972976571, which is matched to 12 digits with $n_0 = 12$ and $n_1 = n_2 = 24$.

4 Domains with several components

We now turn to domains E with $K \ge 2$ connected components $E_1, \dots, E_K$, each with interior; the complement $C \cup \{\infty\}\backslash E$ in the extended complex plane will be multiply connected. For each k, $1 \le k \le K$, we choose a center point c_k in the interior of E_k, and for each corner z_j, $1 \le j \le m$, we let $k(j)$ be the index of the component E_k that z_j belongs to.

We will now need a Laurent polynomial term $q_k(1/(z - c_k))$ associated with each component E_k, whose degree could be denoted by ν_k. A further feature also appears: it is necessary to introduce logarithmic terms in the approximation. The appropriate extension of (2) can be written

$$g(z) = \sum_{k=1}^{K} q_k\Big(\frac{1}{z-c_k}\Big) + \sum_{j=1}^{m} p_j\bigg(\frac{1}{\log\big(\frac{z-z_j}{z-c_{k(j)}}\big) - \frac{1}{2}n_j}\bigg) + \sum_{k=1}^{K} a_k \log(z-c_k), \qquad \sum_{k=1}^{K} a_k = 0 \tag{7}$$

for some real coefficients $\{a_k\}$. The constraint on the sum of the coefficients ensures that no branch point is introduced at $z = \infty$. In computational practice, rather than imposing this constraint explicitly by modification of the linear algebra problem, we impose it implicitly by replacing (7) by

$$g(z) = \sum_{k=1}^{K} q_k\Big(\frac{1}{z-c_k}\Big) + \sum_{j=1}^{m} p_j\bigg(\frac{1}{\log\big(\frac{z-z_j}{z-c_{k(j)}}\big) - \frac{1}{2}n_j}\bigg) + \sum_{k=1}^{K} a_k \log\Big(\frac{z-c_k}{z-c_{k'}}\Big), \tag{8}$$

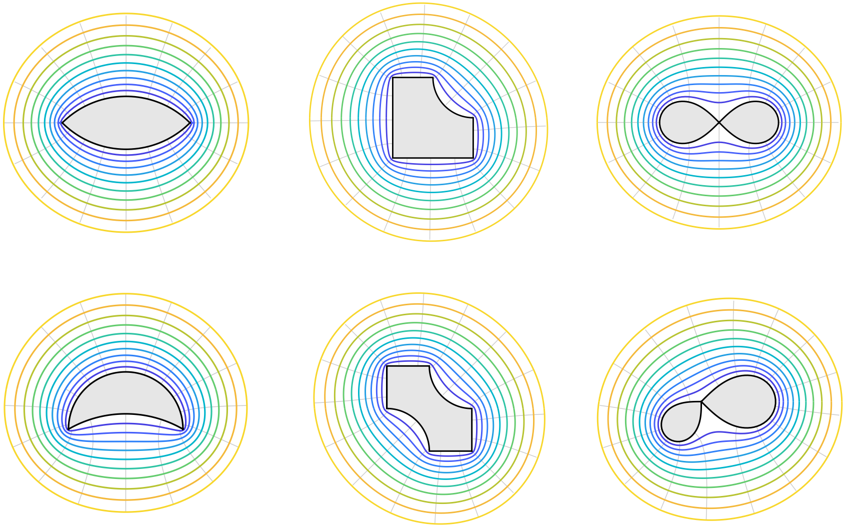

Fig. 2. Six more connected domains with interior. These six solutions to 11–15 digits of accuracy followed by contour plotting took a combined time of about 3 seconds.

where $k' = k(\text{mod}K) + 1$.

A beautiful explanation of the role of logarithmic terms in representation of harmonic functions as real parts of analytic functions has been given by Axler [1]. In a simply-connected domain, the harmonic conjugate is single-valued, and no logarithmic terms are needed. In a multiply-connected domain, however, the harmonic conjugate is multivalued, and a harmonic function can be represented as the real part of a multivalued analytic function that has one logarithmic term per hole. Axler calls this the *logarithmic conjugation theorem,* dating at least to Walsh in 1929. It follows that whereas analytic functions can be approximated by rational functions in multiply connected domains, as shown by Runge in 1885, harmonic functions generally require rational functions plus logarithmic terms, as worked out by Walsh and Keldysh in the 1930s [15, 18]. The function (7) is the analogous formula for reciprocal-log approximations, and has the key property that the imaginary part changes by a constant along any closed contour in $C \cup \{\infty\} \backslash E$ that encloses components of E.

Note that the real parts of (7) and (8) are single-valued, since the coefficients a_k are real, and it is real parts that appear in our least-squares matrices. Specifically, the logarithmic terms introduce K new columns in the matrix A corresponding to samples of $\log(|z - c_k|/|z - c_{k'}|)$.

We find that the log-lightning method works straightforwardly for disconnected domains, again giving near-exponential convergence down to 12 digits of accuracy or so. Figure 3 presents three examples. The first image shows a configuration from [8]: one component of E is a hexagon with corners at -6.5, $-5 \pm 1.5i$, $-5.75 \pm 2.25i$, -8, and the other is a square with corners at 9.5, $8.75 \pm 0.75i$, 8. We place 20 confluent singularities at each of these corners and match the apparently correct value 4.082272420831 with error 3×10^{-11}. (The value reported in [8] is 4.082273.) With 300 points discretizing each side, giving a matrix of size 3000×524, this computation takes 1/6 sec.

The second image, consisting of the unit disk together with two half-disks centered at 3 and $3i$, originates in [18] (see Appendix A). We discretize the circle with 200 points and the other boundary components with 400 points and place Laurent polynomials of degree 15 at the center of each component and 20 singularities at each corner. This gives $\text{cap}(E)$ with error 6×10^{-13} as compared with our best estimate of 2.19699371716970 (which confirms the "best guess" of 2.19699371717 cited

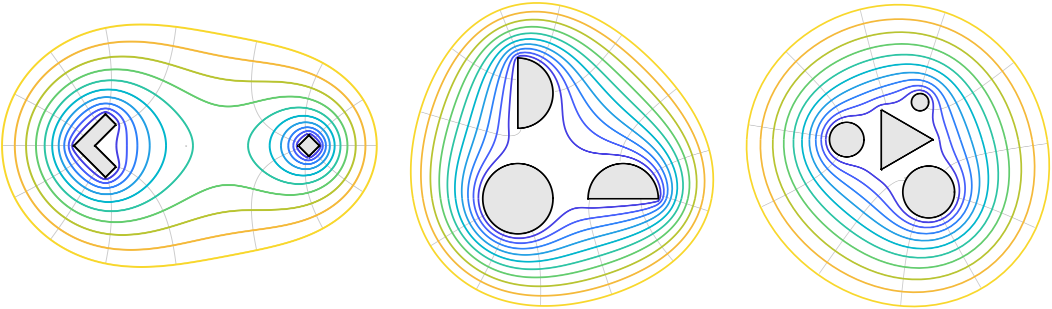

Fig. 3. Disconnected domains with two, three, and four components. To 12 digits, the capacities are 4.08227242083, 2.19699371717, and 1.60724349524. (The images are plotted on different scales, so these numbers are unrelated to the apparent sizes.)

in [17]). This computation with a matrix of dimensions 1800 × 204 takes 1/25 sec., hundreds of times faster than the timing reported in [11].

The third image consists of an equilateral triangle with side length $\sqrt{3}$ and disks centered at $1.25e^{i\pi/3}$, -1.5, and $1.75e^{-i\pi/3}$ of radii 0.25, 0.5, and 0.75, respectively. With 20 confluent poles at each corner and degree 20 Laurent series at each center, the capacity matches the correct value of about 1.60724349524 with error 1.4×10^{-12}. The matrix is of dimensions 1800 × 296, and the computation takes 1/40 secs.

5 Domains defined by arcs

In the opening sentence, we defined E to be a nonempty compact set in the complex plane with no isolated points. As is well known, any such domain has positive capacity, even though its measure as a planar set will be zero if it is composed of one or more more open arcs.

From the point of view of Green's functions and complex analysis, this effect is very natural, and we would like to offer a perhaps unusual interpretation of it. Consider the lens domain E shown as the first image of Figure 2. In the case of a connected domain like this, the level curves of the Green's function can be interpreted as images of concentric circles under a conformal map of the exterior of the unit disk to the exterior of E with ∞ mapping to itself (see e.g. [15]). Suppose now that the half-angle of the lens is reduced from $\pi/4$ to 0, so that the lens collapses to the interval $[-1, 1]$. Clearly nothing essential changes about the conformal map (which is now the Joukowsky transformation). So it is natural that nothing fundamental should change about the capacity; only its value will adjust with the angle. The formula for a lens with chord $[-1, 1]$ and half-angle $\alpha\pi$ is

$$\operatorname{cap}(E) = \frac{1}{2 - 2\alpha}. \tag{9}$$

For $\alpha = 1/2$ we have the unit disk, with capacity 1, and for $\alpha = 0$ we have the slit $[-1, 1]$, with capacity 1/2.

But something startling now emerges: the value $\alpha = 0$ is not a singularity of this formula! The slit is not a boundary case, but an intermediate case between domains whose two sides are separated by a positive amount and domains whose two sides overlap by a positive amount. Configurations with $\alpha < 0$ can be interpreted as "lenses with negative measure." The capacity is positive even though the measure is negative, with $\alpha = -1/2$, for example, corresponding to a "negative unit disk" with $\operatorname{cap}(E) = 1/3$. The conformal map does not change in any essential way, becoming a map onto a domain with two Riemann sheets that happen to overlap.

Mathematically, then, arcs should not be much different from domains with interior. Algorithmically, however, we have a problem. For each corner z_j, the approximation (7) has an order n_j pole near the center point $c_{k(j)}$; the precise location is

$$z = \frac{c_{k(j)} - z_j e^{-n_j/2}}{1 - e^{-n_j/2}}. \tag{10}$$

If E is composed of arcs, then $C \cup \{\infty\} \backslash E$ and ∂E together comprise all of the extended complex plane, so these poles will prevent successful approximation. The same issue arises with lightning as opposed to log-lightning approximations.

Once identified, the problem is readily resolved via local transformations, following an idea introduced by the second author around 2005 (very likely also by others) that has been applied in [6] and [19] for Laplace problems without singularities. To begin the explanation, suppose E contains an analytic arc Γ that runs from $z = -1$ to $z = 1$ and we want to place reciprocal-log terms near both endpoints to approximate possible singularities there. The key idea is to introduce a new variable w defined by an inverse-Joukowsky map,

$$w(z) = z \pm \sqrt{z^2 - 1}, \tag{11}$$

taking care to choose signs for each $z \notin \Gamma$ so that this evaluates to a single branch for all $z \notin \Gamma$. If Γ is the interval $[-1, 1]$ itself, then $w(z)$ ranges over the exterior of the unit disk as z ranges over $C \backslash [-1, 1]$. The crucial observation is that even if Γ is any other arc, $w(z)$ still ranges over a region of the plane that omits a neighborhood of $w = 0$. *The inverse-Joukowsky map opens up arbitrary arcs, not just straight segments.* We can now do reciprocal-log approximation as usual, but in the w variable, taking $w_c = 0$ as the center point. If the whole domain E consists of nothing but Γ, then (2) is replaced by

$$g(z) = p_0\left(\frac{1}{w(z)}\right) + \sum_{j=1}^{2} p_j\left(\frac{1}{\log((w(z) - w_j)/w(z)) - \frac{1}{2}n_j}\right) \tag{12}$$

with $w_1 = -1$ and $w_2 = 1$. More generally, E will contain more than just one arc, and we use a different local variable w for each one, with each z in (8) being replaced by an appropriate $w(z)$ depending on which part of E it belongs to. By the procedure just described we can apply the log-lightning idea (or lightning rational approximations) to general domains that contain arcs or indeed portions of “negative measure” as discussed above. This is related to the procedure called “opening up” in [11], which however involves a global conformal map rather than local ones, and may answer a question posed in the last sentence of that paper.

Figure 4 shows Green’s functions computed in this manner for three domains composed of arcs. On the left, the three-segment domain has corners at ± 1 and $\pm 1 \mp 0.8i$ (see Appendix B). It is treated numerically by six intervals, since each segment has two sides. With 500 exponentially clustered sample points on each interval, Laurent polynomials of degree 8 and 16 reciprocal-log poles in the w variable associated with each corner, we get a matrix of dimension 3000×258 and a capacity computation in 0.08 sec. that matches the correct value 1.16330695044503 with error 3×10^{-14}.

The middle image of Figure 4 shows a star with 5 points, whose capacity is known analytically as $2^{-2/5}$. With Laurent polynomials of degree 12 and 20 poles near each corner, we get a matrix of dimensions 4000×340 and error 4×10^{-13} in $1/5$ sec. Here as throughout this article, except in the the example of the square at the beginning, no advantage has been taken of symmetry.

The right image of the first shows a circular arrow formed from the upper half of the unit circle together with segments of length 0.8 at angles $\pi/3$ and $5\pi/6$ from the positive real axis. With a Laurent polynomial of degree 40 and 15 singularities near each endpoint we match the capacity

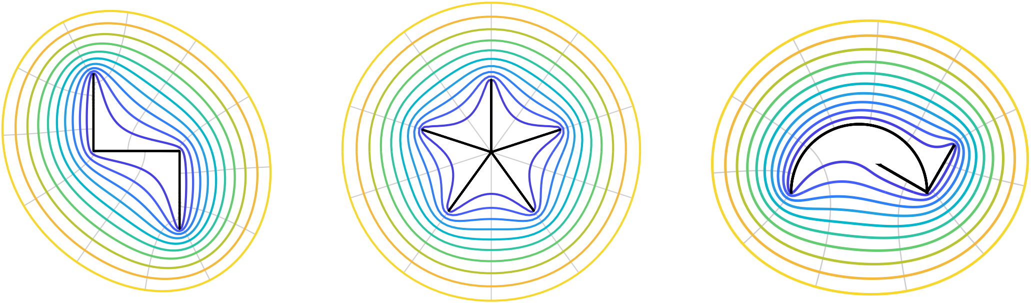

Fig. 4. Green's functions for three domains composed of arcs, computed by reciprocal-log approximation with a change of variables of type (11) applied locally on each arc. To 12 digits, the capacities are 1.16330695044, 0.757858283255, and 0.798004019450.

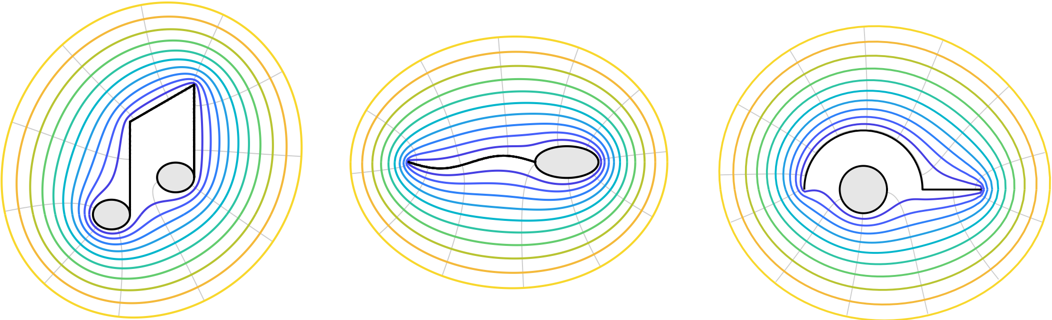

Fig. 5. Three hybrid domains combining open arcs and portions with interior. The capacities are 3.0508716, 0.82598425, and 0.970118179695.

of about 0.798004019450 with an error 2.6×10^{-9}. The matrix has dimension 3600×242 and the computation takes $1/8$ sec.

Our final figure, Figure 5, shows three domains that combine arcs, portions with interior, and multiple connectivity. The eighth notes (quavers) are defined by ellipses of semiaxis lengths 1 and 0.8 centered at 0 and $3.5 + 2i$ with stems of length 5; the capacity to 6 digits is 3.0508716. The spermatozoon has a tail defined by $z = x + 0.1i \sin(\pi x)$ for $1 \le x \le 1$ and its head is an ellipse of semiaxes 0.5 and 0.25 centered at $z = 1.5$; the capacity is 0.82598425. The baseball cap (or is it a wayward question mark?) consists of the interval $[1, 2]$, the upper half of the unit circle, and the disk about 0 of radius 0.4; it has capacity 0.970118179695.

6 Discussion

The paper [12] introducing the idea of log-lightning Laplace solutions gives just one page to this topic at the end of an extensive discussion of the underlying approximation theory, with theorems justifying our claim of almost-exponential convergence. The present report thus represents the first serious investigation of the method. We have found that it has worked with no surprises, successfully computing Green's functions and capacities for all the domains we have considered, giving approximately exponential convergence of $\mathrm{cap}(E)$ down to 10-15 digits of accuracy. Each computed capacity we have reported by its decimal expansion comes from a consensus of a few dozen numerical experiments with varying sets of clustered sample points and approximation degrees, giving us reasonable confidence in each case that all digits are correct except perhaps the final one listed.

From a practical point of view, we do not claim that this is a major advance over the earlier lightning method with its root-exponential convergence, since both can typically compute solutions to 8–10 digit accuracy in comparable computer times and few applications would require more. Conceptually, however, the log-lightning method seems potentially an important advance. It effects a more or less complete decoupling between the global scale of a domain boundary and a local scale near each singularity, and this may contribute to a deeper understanding of certain multiscale aspects of partial differential equations.

For the academic challenge of computing capacities to, say, 100 digits, the log-lightning method should be superb. All that is needed is an extended precision environment that is good at linear algebra, since significant matrix computations are involved. It is possible that difficulties may arise here, since the MATLAB backslash operator we have relied on has a good deal of engineering under the hood, whose details we do not know. In particular, it reliably returns low-norm (hence in some sense regularized) solution vectors where possible for least-squares problems that are highly ill-conditioned or even rank-deficient. Taking advantage of this, we have made no effort to avoid rank-deficient matrices, which arise regularly since the columns may include several copies of the constant vector coming from the zero-degree terms of different polynomials. If, as we hope, others implement log-lightning Laplace solutions in extended precision, this may be an issue to pay attention to.

Of course, a great deal is already known about Green's functions and their computation. Concerning the conceptual relationships between simply and multiply connected domains, we note the recent book by Crowdy summarizing twenty years of research related to the unifying tool known as the prime function [7]. Concerning numerical computation, the most advanced methods are based on integral equations, as presented in [13]. Contributions focused specifically on the computation of capacity include a paper of Liesen, Sète, and Nasser and a succession of works by Rostand, Ransford, and Rajon [14, 16, 17, 18]. Of these the paper [18] is closest in spirit to the present contribution, being based on rational approximations determined by least-squares fitting on the boundary.

Like the lightning method, the log-lightning method has some very attractive features. It finds a solution very quickly, and it could be even faster if software is developed like the lightning code [20] with the number of singularities at each corner and the degrees of polynomial terms determined adaptively. It gives a global and therefore perfectly smooth representation of the solution, which can be evaluated in a matter of microseconds per point. Harmonic conjugates come for free (hence the Hilbert transform or Dirichlet-to-Neumann map in related settings). A posteriori error guarantees can be computed with the maximum principle. As discussed in [12], there are also interesting possibilities for calculations on Riemann surfaces, which we have not investigated here.

References

[1] Sheldon Axler, Harmonic functions from a complex analysis viewpoint, *Amer. Math. Monthly* **93** (1986), 246–258.

[2] Peter J. Baddoo, Lightning solvers for potential flows, *Fluids* **5** (2020), 227.

[3] Pablo D. Brubeck, Yuji Nakatsukasa, and Lloyd N. Trefethen, Vandermonde with Arnoldi, *SIAM Rev.* **63** (2021), 405–415.

[4] Pablo D. Brubeck and Lloyd N. Trefethen, Lightning Stokes solver, *SIAM J. Sci. Comput.*, submitted.

[5] W. J. Cody, G. Meinardus, and R. S. Varga, Chebyshev rational approximations to e^{-x} on $[0, +\infty)$ and applications to heat-conduction problems, *J. Approx. Th.* **2** (1969), 50–65.

[6] Stephen M. Cox and Matthew D. Finn, Two-dimensional Stokes flow driven by elliptical paddles, *Phys. Fluids* **19** (2007), 113102.

[7] Darren Crowdy, *Solving Problems in Multiply Connected Domains,* SIAM (2020).

[8] Mark Embree and Lloyd N. Trefethen, Green's functions for multiply connected domains via conformal mapping, *SIAM Rev.* **41** (1999), 745–761.

[9] Abinand Gopal and Lloyd N. Trefethen, Solving Laplace problems with corner singularities via rational functions, *SIAM J. Numer. Anal.* **57** (2019), 2074–2094.

[10] Abinand Gopal and Lloyd N. Trefethen, New Laplace and Helmholtz solvers, *Proc. Nat. Acad. Sci.* **116** (2019), 10223.

[11] Jörg Liesen, Olivier Sète, and Mohamed M. S. Nasser, Fast and accurate computation of the logarithmic capacity of compact sets, *Comp. Meth. Funct. Th.* **17** (2017), 689–713.

[12] Yuji Nakatsukasa and Lloyd N. Trefethen, Reciprocal-log approximation and planar PDE solvers, *SIAM J. Numer. Anal.*, to appear.

[13] Manas Rachh and L. Greengard, Integral equation methods for elastance and mobility problems in two dimensions, *SIAM J. Numer. Anal.* **54** (2016), 2889–2909.

[14] Quentin Rajon, Thomas Ransford, and Jérémie Rostand, Computation of capacity via quadratic programming, *J. math. pures et appliq.* **94** (2010), 398–413.

[15] Thomas Ransford, *Potential Theory in the Complex Plane*, Cambridge, 1995.

[16] Thomas Ransford, Computation of logarithmic capacity, *Comp. Meth. Funct. Th.* **10** (2011), 555–578.

[17] Thomas Ransford and Jérémie Rostand, Computation of capacity, *Math. Comput.* **76** (2007), 1499–1520.

[18] Jérémie Rostand, Computing logarithmic capacity with linear programming, *Experimental Math.* **6** (1997), 221–238.

[19] Lloyd N. Trefethen, Series solution of Laplace problems, *ANZIAM J.* **60** (2018), 1–26.

[20] Lloyd N. Trefethen, Lightning Laplace software, `http://people.maths.ox.ac.uk/trefethen/lightning` (2020).

A MATLAB code for three-component domain

As a first template showing log-lightning computations, we list here a code to compute the Green's function and capacity for the three-component domain of Figure 3.

```
%% Geometry
M = 400;
s = tanh(linspace(-10,10,M)');
Z = [s; 1i*exp(.5i*pi*s)];
Z1 = Z+3; c1 = 3+.5i; Z2 = Z/1i+3i; c2 = .5+3i;
Z3 = exp(2i*pi*(1:M/2)'/(M/2)); Z = [Z1; Z2; Z3];
z1 = 2; z2 = 4; z3 = 2i; z4 = 4i;

%% Matrix problem
F = log(abs(Z));
nP = 20; s0 = nP/2;
[H1,L1] = VAorthog(1./(s0-log(1-(z1-c1)./(Z-c1))),nP);
[H2,L2] = VAorthog(1./(s0-log(1-(z2-c1)./(Z-c1))),nP);
[H3,L3] = VAorthog(1./(s0-log(1-(z3-c2)./(Z-c2))),nP);
[H4,L4] = VAorthog(1./(s0-log(1-(z4-c2)./(Z-c2))),nP);
nL = 15;
L = [Z.^(-1:-1:-nL) L1 L2 L3 L4];
A = [log(abs(1./(1-c1./Z))) log(abs((1-c1./Z))./(1-c2./Z)) ...
     log(abs(1-c2./Z)) real(L) 0*Z 0*Z 0*Z -imag(L)];
warning off, cc = A\F;

%% Function handles for solution
c = reshape(cc,[],2)*[1; 1i];
g = @(z) reshape([log(1./(1-c1./z(:))) log((1-c1./z(:))./(1-c2./z(:))) ...
          log((1-c2./z(:))) z(:).^(-1:-1:-nL) ...
          VAeval(1./(s0-log(1-(2 -c1)./(z(:)-c1))),H1) ...
          VAeval(1./(s0-log(1-(4 -c1)./(z(:)-c1))),H2) ...
          VAeval(1./(s0-log(1-(2i-c2)./(z(:)-c2))),H3) ...
          VAeval(1./(s0-log(1-(4i-c2)./(z(:)-c2))),H4)]*c,size(z));
```

```
u = @(z) real(g(z)); v = @(z) imag(g(z));
cap = exp(u(inf));
Gcmplx = @(z) log(z) - g(z);
G = @(z) real(Gcmplx(z)); H = @(z) imag(Gcmplx(z));

%% Contour plot
LW = 'LineWidth'; PO = 'position';
axes(PO,[.03 .36 .57 .57])
x = linspace(-4,6,250); y = linspace(-4,6,250);
[xx,yy] = meshgrid(x,y); zz = xx+1i*yy;
body = @(z) fill(real(z),imag(z),[.9 .9 .9],LW,1.5);
body(Z1), hold on, body(Z2), body(Z3)
inpolygonc = @(z,w) inpolygon(real(z),imag(z),real(w),imag(w));
ii = inpolygonc(zz,Z1) + inpolygonc(zz,Z2) + inpolygonc(zz,Z3);
GG = G(zz); GG(ii>0) = NaN;
contour(x,y,GG,(.1:.1:1)*2/3,LW,1.5)
levelsH = mod(linspace(0,2*pi,15)-.8*pi/8,2*pi)-pi; levelsH(end)=[];
HH = H(zz); HH(ii>0) = NaN; HH(G(zz)>2/3) = NaN;
HH(abs(zz-1.3-1.3i)<1.3) = NaN;
contour(x,y,HH,levelsH,'Color',.7*[1 1 1],LW,.75), hold off
axis([-4 6 -4 6]), axis equal, caxis([0,2/3*1.1])
title(sprintf('cap(E) = %12.9f', cap))

%% Error plot on boundary
axes(PO,[.62 .54 .34 .33])
err = abs(A*cc-F);
semilogy(err), axis([0 length(err) 1e-15 1]), grid on
title('error at boundary points')
```

B MATLAB code for three-segment domain

As a second template, illustrating the use of Joukowsky transformations, here is a code to compute the Green's function and capacity for the first domain of Figure 4.

```
%% Geometry
M = 200;
s = tanh(linspace(-15,15,M));
L = 1.8i;
su = s + 10i*eps; sl = s - 10i*eps;
Z = [-1+L*(sl+1)/2, su, 1-L*(su+1)/2 -1+L*(su+1)/2, sl, 1-L*(sl+1)/2].';
z1 = -1+L; z2 = -1; z3 = 1; z4 = 1-L;

%% Joukowsky variable with branch cut [a,b]
w = @(z,a,b) z - (a+b)/2 + exp((log(exp(-1i*angle(a-b))*(z-a)) ...
                              + log(exp(-1i*angle(a-b))*(z-b)))/2 ...
                              + 1i*angle(a-b));

%% Matrix problem
nL = 0; nP = 20; s0 = nP/2;
lc = @(z,p,a,b) 1./(s0-log(1-w(p,a,b)./w(z,a,b)));
[H1a,P1 ] = VAorthog(1./w(Z,z1,z2),nL);      % segment 1
```

```
[H1b,L1b] = VAorthog(lc(Z,z1,z1,z2),nP);
[H1c,L1c] = VAorthog(lc(Z,z2,z1,z2),nP);
[H2a,P2 ] = VAorthog(1./w(Z,z2,z3),nL);      % segment 2
[H2b,L2b] = VAorthog(lc(Z,z2,z2,z3),nP);
[H2c,L2c] = VAorthog(lc(Z,z3,z2,z3),nP);
[H3a,P3 ] = VAorthog(1./w(Z,z3,z4),nL);      % segment 3
[H3b,L3b] = VAorthog(lc(Z,z3,z3,z4),nP);
[H3c,L3c] = VAorthog(lc(Z,z4,z3,z4),nP);
A = [P1 L1b L1c P2 L2b L2c P3 L3b L3c];
A = [real(A) -imag(A)];
source = @(z) log(w(z,-1,1)/2);
F = real(source(Z));
cc = A\F;

%% Function handles for solution
c = reshape(cc,[],2)*[1; 1i];
g = @(z) reshape([VAeval(1./w(z(:),z1,z2),H1a),...
   VAeval(lc(z(:),z1,z1,z2),H1b)...
   VAeval(lc(z(:),z2,z1,z2),H1c)...
   VAeval(1./w(z(:),z2,z3),H2a),...
   VAeval(lc(z(:),z2,z2,z3),H2b)...
   VAeval(lc(z(:),z3,z2,z3),H2c)...
   VAeval(1./w(z(:),z3,z4),H3a)...
   VAeval(lc(z(:),z3,z3,z4),H3b)...
   VAeval(lc(z(:),z4,z3,z4),H3c)]*c,size(z));
u = @(z) real(g(z)); v = @(z) imag(g(z));
cap = exp(u(inf));
Gcmplx = @(z) source(z) - g(z);
G = @(z) real(Gcmplx(z)); H = @(z) imag(Gcmplx(z));

%% Contour plot
LW = 'LineWidth'; PO = 'position';
axes(PO,[.03 .36 .57 .57])
x = linspace(-4,6,200); y = linspace(-4,6,200);
[xx,yy] = meshgrid(x,y); zz = xx+1i*yy;
HH = H(zz); HH(G(zz)>1) = NaN;
HH(real(zz)<-1 & abs(imag(zz))<0.1) = NaN;
levelsH = linspace(min(min(HH)),max(max(HH)),14); levelsH([1,end]) = [];
contour(x,y,HH,levelsH,'Color',.7*[1 1 1],LW,.75), hold on
contour(x,y,G(zz),.1:.1:1,LW,1.5), caxis([0 1.1])
plot([z1 z2 z3 z4],'k',LW,1.5), axis([-4 4 -4 4]), axis square, hold on
title(sprintf('cap(E) = %12.9f', cap)), hold off

%% Error plot on boundary
axes(PO,[.62 .54 .34 .33])
err = abs(A*cc-F);
semilogy(err), axis([0 length(err) 1e-15 1]), grid on
title('error at boundary points')
```