SIAM J. SCI. COMPUT.
Vol. 25, No. 3, pp. 1042–1065

A MULTIPOLE METHOD FOR SCHWARZ–CHRISTOFFEL MAPPING OF POLYGONS WITH THOUSANDS OF SIDES*

LEHEL BANJAI† AND L. N. TREFETHEN†

Abstract. A method is presented for the computation of Schwarz–Christoffel maps to polygons with tens of thousands of vertices. Previously published algorithms have CPU time estimates of the order $O(N^3)$ for the computation of a conformal map of a polygon with N vertices. This has been reduced to $O(N \log N)$ by the use of the fast multipole method and Davis's method for solving the parameter problem. The method is illustrated by a number of examples, the largest of which has $N \approx 2 \times 10^5$.

Key words. conformal mapping, fast multipole method, Schwarz–Christoffel mapping, Koch snowflake

AMS subject classifications. 30C30, 65E05, 70F10

DOI. 10.1137/S1064827502411675

1. Introduction.

Conformal mapping has been long recognized as a useful tool in the solution of partial differential equations. Many numerical methods have been developed for the construction of conformal maps between a canonical domain, e.g., the unit disk or the half-plane, and a physical domain, a more or less arbitrary simply connected domain Ω. Some requirements are normally placed on the physical domain to ensure the practicality of the method: some methods require $\partial\Omega$ to be smooth, others require it to be piecewise smooth, etc. The method presented here assumes that $\partial\Omega$ is a polygon; it is able to deal with a very large number of vertices, making it applicable also in the cases mentioned above. For this special case a semiexplicit formula has been known for many years, the Schwarz–Christoffel (SC) formula, named after its independent discoverers E. B. Christoffel (1867) and H. A. Schwarz (1869).

Around 1980 the second author developed a robust computer package called SCPACK, written in FORTRAN, for the computation of SC maps [35, 36]. The SC toolbox for MATLAB by Driscoll is a descendent of SCPACK with wider capabilities and is able to compute mappings to polygons with a hundred or so vertices [8]. Detailed descriptions of algorithms implemented in these packages and of many applications of SC mapping can be found in the recent book [9]. In both these implementations the cost of evaluating the conformal map at a single point, once all the unknowns in the SC formula have been computed, is $O(N)$, where N is the number of vertices of the polygon, and the cost of computing these unknowns, using a quasi-Newton iteration, is $O(N^3)$. In this paper these times have been reduced to $O(\log N)$ and $O(N \log N)$, respectively (these figures represent typical behavior for most polygons, not guaranteed behavior for all polygons). The improvements in the time complexity are due to the use of the fast multipole method (FMM) developed by Carrier, Greengard, and Rokhlin [4, 16, 34] and the use of a simple iteration for finding the unknown parameters in the SC formula due to Davis [7]. At present our algorithm is restricted to bounded polygons, although extensions to the unbounded case could certainly be considered.

*Received by the editors July 22, 2002; accepted for publication (in revised form) April 25, 2003; published electronically November 21, 2003.
http://www.siam.org/journals/sisc/25-3/41167.html
†Computing Laboratory, Oxford University, Wolfson Building, Parks Road, Oxford OX1 3QD, UK (lehelb@comlab.ox.ac.uk, LNT@comlab.ox.ac.uk).

This is not the first time that the FMM has been applied to numerical conformal mapping. O'Donnell and Rokhlin [29] modified an algorithm [25, 37] for computing conformal maps of smooth domains via the Szegö kernel. Each iteration of their process requires that an $N \times N$ matrix, where N is the number of discretization points of the boundary, be applied to a vector, which is normally an $O(N^2)$ procedure, but by exploiting the FMM and the particular structure of the matrix they were able to reduce this to $O(N)$.

A generalized SC transformation was developed by Floryan and Zemach [13]. In particular they consider conformal maps of infinite polygons with periodic boundary. Their method also allows for curved segments of the boundary without the need to approximate these segments with a large number of sides.

Another notable predecessor to our work is the celebrated computation of millions of zeros of the Riemann zeta function near $z = \frac{1}{2} + i10^{20}$ by Odlyzko, using an algorithm of Schönhage [30, 31, 32], which formed the basis of a careful comparison of the spacings of Riemann zeros with those of eigenvalues of random matrices. This algorithm was not precisely a multipole algorithm, but it made use of similar ideas of recursive expansion in the complex plane.

2. Fast multipole method for SC mapping. In this paper we consider only conformal maps from the unit disk to the interior of a polygon; exterior maps can be treated by similar methods. The SC formula for the interior map of the unit disk to a polygon is

$$f(z) = A + C \int^z \prod_{k=1}^{N} (\zeta - z_k)^{\alpha_k - 1} d\zeta, \tag{2.1}$$

where $\alpha_k \pi$ are the interior angles of the polygon, z_k are the *prevertices*, and A and C are constants. The lower integration limit is left unspecified since it affects only the constant A. We want to be able to evaluate this function at M points in time $O(M \log N + N)$. To do this we consider the logarithm of the integrand and use a fast multipole-type method to achieve our goal. We will consider sums of the type

$$G(z) = \sum_{j=1}^{L} \beta_{k_j} \log(z - z_{k_j}), \tag{2.2}$$

where $\beta_{k_j} = \alpha_{k_j} - 1$ and $L \leq N$. In the FMM such sums are considered for clustered groups of z_{k_j} so that, for z distant to the cluster, $G(z)$ could be accurately evaluated using compressed information about the prevertices. Hence the sums we consider most often consist only of a subset of all the N prevertices.

First, we state and prove a theorem which allows us to apply the FMM to the problem at hand. Once we have this result we can show how the FMM can be used to evaluate (2.2) efficiently.

2.1. Expansion of the integrand and error bounds. In this section $\mathcal{T}$ denotes the unit circle $|z| = 1$ and $D(z_0, r)$ the closed disk centered at z_0 of radius r.

LEMMA 2.1. *Let $z_1 \in \mathcal{T}$ and $z, z_0 \in \mathbb{C}$ be such that $|\frac{z_1 - z_0}{z - z_0}| < 1$. Then*

$$\log(z - z_1) = \log(z - z_0) - \sum_{k=1}^{\infty} \frac{1}{k} \left(\frac{z_1 - z_0}{z - z_0} \right)^k. \tag{2.3}$$

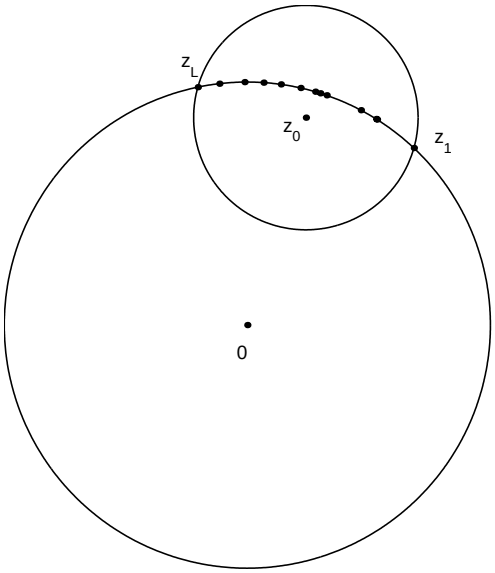


FIG. 1. *The set-up for Lemma* 2.2.

Proof. Since $\log(z - z_1) = \log(z - z_0) + \log(1 - \frac{z_1 - z_0}{z - z_0})$, the lemma follows from the expansion

$$\log(1 - w) = -\sum_{k=1}^{\infty} \frac{w^k}{k},$$

which is valid for $|w| < 1$. ∎

LEMMA 2.2. *Let $\beta_i \in [-1, 1]$, $i = 1, \ldots, L$, be such that for all j with $1 \leq j \leq L$, $|\sum_{i=1}^{j} \beta_i| \leq C$, where C is a constant. Let $z_1, z_2, \ldots, z_L$ be points on $\mathcal{T}$ in counterclockwise order such that $\arg(z_L) - \arg(z_1) < \pi$. Also let $z_0 = \frac{z_1 + z_L}{2}$ and $r = \frac{|z_1 - z_L|}{2}$ (see Figure* 1*). Then for any $k \geq 1$,*

$$\left| \sum_{i=1}^{L} \beta_i (z_i - z_0)^k \right| \leq (1 + k\pi) C r^k. \tag{2.4}$$

Proof. Let $\nu_i = z_i - z_0$. This implies $|\nu_i| \leq r$ and $\arg(\nu_L) - \arg(\nu_1) < \pi$. Rotate the x- and y-axes so that all the ν_i's are in the first two quadrants of $D(0, r)$ (see Figure 2). This will not change the size of the sum we are trying to estimate. Notice that

$$\sum_{j=1}^{L-1} |\nu_{j+1} - \nu_j| \leq r\pi. \tag{2.5}$$

Define sequences ω, b, and w by

$$\omega_i = \nu_i^k, \quad b_i = \sum_{j=1}^{i} \beta_j, \quad w_i = \omega_i - \omega_{i-1},$$

for $i = 1, \ldots, L$, with boundary cases $b_0 = w_1 = 0$.

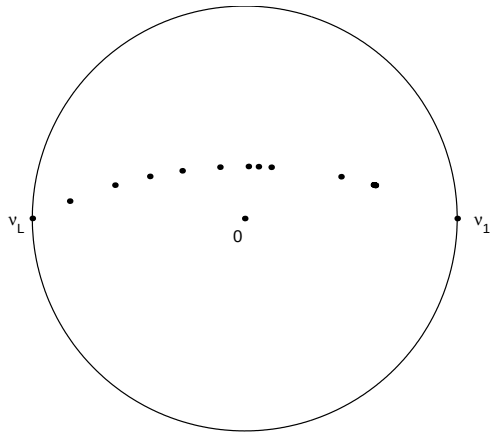


FIG. 2. *Circle centered at* $(0,0)$ *with radius* r. *Points* ω_i *are shown for* $k=1$.

Two facts to notice are that

$$||b||_\infty \le C \tag{2.6}$$

and

$$\begin{aligned} ||w||_1 &= \sum_{j=1}^{L-1} |\omega_{j+1}-\omega_j| = \sum_{j=1}^{L-1} |\nu_{j+1}^k - \nu_j^k| \\ &= \sum_{j=1}^{L-1} |\nu_{j+1}-\nu_j| \left| \sum_{i=1}^{k} \nu_{j+1}^{k-i}\nu_j^{i-1} \right|, \end{aligned}$$

and hence by (2.5) and the fact $|\nu_j| \le r$,

$$\begin{aligned} ||w||_1 &\le kr^{k-1} \sum_{j=1}^{L-1} |\nu_{j+1}-\nu_j| \\ &\le kr^k\pi. \end{aligned} \tag{2.7}$$

We now follow an argument of summation by parts. Since

$$b_i\omega_i - b_{i-1}\omega_{i-1} = (b_i - b_{i-1})\omega_i + (\omega_i - \omega_{i-1})b_{i-1} = \beta_i\omega_i + b_{i-1}w_i,$$

we have

$$\beta_i\omega_i = b_i\omega_i - b_{i-1}\omega_{i-1} - b_{i-1}w_i.$$

Adding up a series of such terms, we find that the cross-terms cancel and we are left with

$$\sum_{i=1}^{L} \beta_i\omega_i = b_L\omega_L - \sum_{i=1}^{L} b_{i-1}w_i. \tag{2.8}$$

Using (2.6), (2.7), the above expression, and the fact $|\omega_L| = r^k$, by Hölder's inequality we obtain the required bound

$$\left| \sum_{i=1}^{L} \beta_i (z_i - z_0)^k \right| \le ||b||_\infty (r^k + ||w||_1) \le (1+k\pi)Cr^k. \tag{2.9}$$

□

The following is the main result. It shows how to construct the Laurent expansion, plus a term involving a logarithm, of the logarithm of the SC integrand. Also it provides an estimate of the error when this expansion is approximated by a truncated expansion of p terms.

THEOREM 2.3. *Let z_i, β_i, r, $i = 1, \ldots, L$, and z_0 be as in Lemma 2.2. Then for all $z \in \mathbb{C}$ such that $|z - z_0| > r$, the function $G(z) = \log(\prod_{i=1}^{L}(z - z_i)^{\beta_i})$ can be expressed as*

$$G(z) = a_0 \log(z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k}, \tag{2.10}$$

where

$$a_0 = \sum_{i=1}^{L} \beta_i \ \text{ and } \ a_k = -\sum_{i=1}^{L} \beta_i \frac{(z_i - z_0)^k}{k}. \tag{2.11}$$

Furthermore, for any $p \geq 1$,

$$\left| G(z) - a_0 \log(z - z_0) - \sum_{k=1}^{p} \frac{a_k}{(z - z_0)^k} \right| \leq \left(\frac{A}{c - 1} \right) c^{-p}, \tag{2.12}$$

where

$$A = (1 + \pi)C \ \text{ and } \ c = \frac{|z - z_0|}{r}. \tag{2.13}$$

Proof. The form of (2.10) is obtained from Lemma 2.1 and the fact that

$$G(z) = \sum_{i=1}^{L} \beta_i \log(z - z_i).$$

To obtain the error bound (2.12) we proceed as follows:

$$\begin{aligned} \left| G(z) - a_0 \log(z - z_0) - \sum_{k=1}^{p} \frac{a_k}{(z - z_0)^k} \right| &= \left| \sum_{k=p+1}^{\infty} \frac{a_k}{(z - z_0)^k} \right| \\ &= \left| \sum_{k=p+1}^{\infty} \sum_{i=1}^{L} \frac{\beta_i}{k} \left(\frac{z_i - z_0}{z - z_0} \right)^k \right| \\ &\leq \sum_{k=p+1}^{\infty} \frac{1}{k|z - z_0|^k} \left| \sum_{i=1}^{L} \beta_i (z_i - z_0)^k \right|. \end{aligned} \tag{2.14}$$

Then by Lemma 2.2 we conclude that

$$\begin{aligned} \left| \sum_{k=p+1}^{\infty} \frac{a_k}{(z - z_0)^k} \right| &\leq \sum_{k=p+1}^{\infty} \frac{(1 + k\pi)C}{k} \left(\frac{r}{|z - z_0|} \right)^k \\ &\leq \sum_{k=p+1}^{\infty} (1 + \pi)C \left(\frac{r}{|z - z_0|} \right)^k \\ &= A \sum_{k=p+1}^{\infty} c^{-k} \\ &= \left(\frac{A}{c - 1} \right) c^{-p}. \qquad \square \end{aligned}$$

Theorem 2.3 ensures that the FMM can be applied for efficient evaluation of SC maps with a large number of prevertices. When implementing these results one needs to be careful that the branch cuts of the logarithms involved point away from the unit disk.

Note that Lemma 2.2 could be proved much more easily if we were satisfied with the estimate

$$\left| \sum_{i=1}^{L} \beta_i (z_i - z_0)^k \right| \le \sum_{i=1}^{L} |\beta_i| r^k \le L r^k.$$

This error bound is fine if L is not much larger than p; see (2.14). For example, if $p = L + 1$, the truncation error in Theorem 2.3 would be bounded by a simple expression:

$$error \ \le \sum_{k=p+1}^{\infty} \frac{r^k}{|z - z_0|^k} = \left(\frac{1}{c-1} \right) c^{-p}.$$

For $L \gg p$, however, results of Theorem 2.3 as written have to be used. Still, for some polygons both methods give an error bound that depends on L. This happens when a long sequence of parameters β_i have the same sign. For example, this can happen for a spiral polygon, or indeed for the Koch snowflake. These considerations are especially necessary if in the future we intend to consider generalizations to polygons with an infinite number of prevertices.

To see what advantage can be gained from expressing the effect of prevertices as a truncated expansion, let us consider a simple example. Suppose we are in the situation of Theorem 2.3 and we want to evaluate the integrand at M points $y_1, \ldots, y_M$ such that $|y_i - z_0| > r$. To do this directly would take $O(LM)$ operations, but to form the multipole expansion and then evaluate it M times takes $O(Lp + Mp)$ operations, which is a big savings if M and L are much greater than p.

2.2. Modified FMM algorithm. In our algorithm the prevertices are all positioned on the unit disk, and the function is evaluated only inside the disk, so the computational cell is the unit circle instead of a square as in the standard FMM algorithm. We could ignore this fact and just place the unit disk inside a square computational cell and use the adaptive version of the FMM. Still, using a nonstandard shape of the computational cell and choosing an appropriate division result in a simpler and faster code.

What we have done is developed an adaptive procedure for the subdivision of the computational cell into cells of different shapes and into different levels of refinement. At level 0 there is only one cell; at level 1 the computational cell is subdivided into 9 cells: $b_0, b_1, \ldots, b_8$, where $b_0 = \{z : |z| \le 0.3742\}$ and $b_k = \{re^{i\theta} : 0.3742 < r \le 1,\ 2\pi(k-1)/8 < \theta \le 2\pi k/8\}$ for $k = 1, 2, \ldots, 8$; see Figure 4. The reason for choosing this particular subdivision will be explained later. Going from level l to level $l + 1$, only cells containing more than some fixed number s of prevertices are subdivided. Hence the circular cell b_0 never gets subdivided.

A general nonempty cell is bounded by two straight lines whose continuations intersect at the origin and two circular arcs centered at the origin of which one is at distance 1 from the origin. A cell like this is subdivided into four cells by a straight line connecting the midpoints of the circular arcs and a circular arc centered at the origin and connecting the midpoints of the straight sides (see Figure 3). For a complete

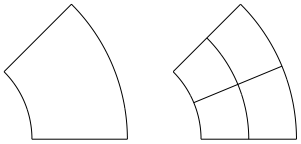


FIG. 3. *Subdivision of a parent cell.*

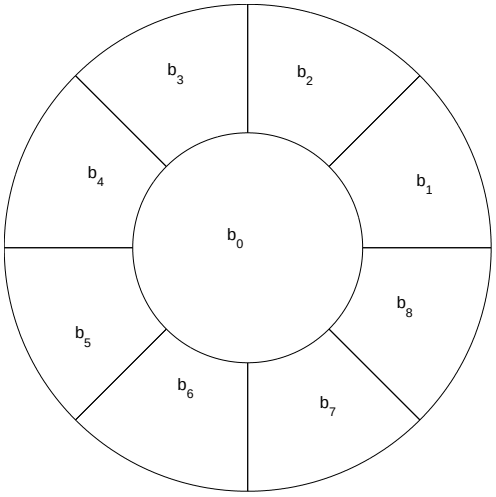


FIG. 4. *Cells* $b_0, \ldots, b_8$ *at level one.*

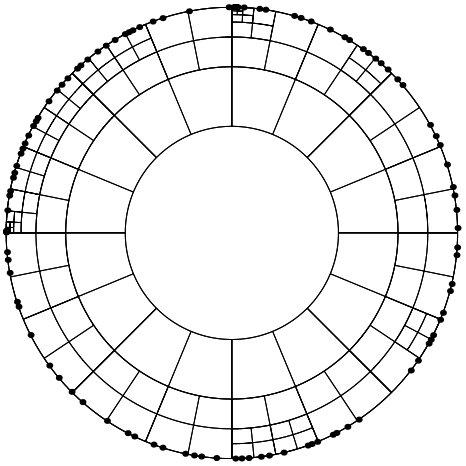


FIG. 5. *Complete adaptive mesh with* $s = 3$.

adaptive mesh of the unit disk, see Figure 5. We say that two cells at the same level of refinement are *well separated* if they are not adjacent. For a cell b we define the *center* z_b to be the point equally spaced from all the corners of the cell and the *radius* r_b to be the distance from the center to the corners.

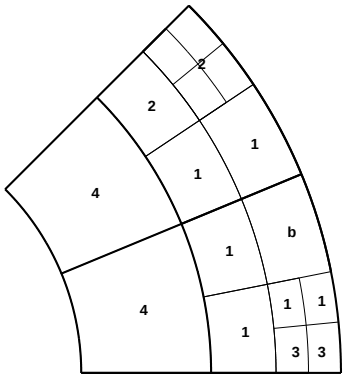


FIG. 6. *A subset of the computational cell is shown. Each cell is labeled according to which list associated with the cell b it belongs to.*

2.2.1. Notation. Before stating the algorithm we introduce several definitions. These definitions, with only a few changes, are taken from the work of Carrier, Greengard, and Rokhlin [4].

- For any subset A of the computational cell, $T(A)$ denotes the set of prevertices contained in A.
- A *child cell* is a cell resulting from a subdivision of a cell.
- A cell that has been subdivided is called a *parent cell*. Otherwise the cell is said to be *childless*.
- *Colleagues* are adjacent cells at the same level.

With each cell b we associate four different lists of cells:

1. U_b contains b and all childless cells adjacent to b.
2. V_b contains all the children of b's parents that are well separated from b (note this can only happen at level 1) and the children of the colleagues of b's parents that are well separated from b.
3. W_b contains all the descendants of b's colleagues whose parents are adjacent to b but who are not adjacent to b themselves.
4. X_b contains all cells c such that $b \in W_c$.

For a visual description of these lists, see Figure 6.

Prevertex z_k is said to be *distant* from a cell b if $z_k \notin T(U_b) \cup T(W_b)$. Also we define four types of expansions associated with a cell b:

- Φ_b denotes the p-term multipole expansion due to $T(b)$.
- Ψ_b denotes the p-term local (Taylor) expansion, valid inside the cell b, representing the effect of distant prevertices.
- Γ_b denotes the p-term local expansion due to all the prevertices in $T(V_b)$.
- Δ_b denotes the p-term local expansion due to all the prevertices in $T(X_b)$.

2.2.2. Translation operators. To complete the mathematical apparatus needed for the FMM we need to define three translation operators. We give only an informal description here, together with error bounds. The details can be found in [15].

- T_1 is a translation operator that shifts the center of a multipole expansion. There is no loss of precision due to this operation. The error is the same as that of the initial truncated expansion.
- T_2 converts a multipole expansion Φ_b into a Taylor series valid in a cell b_1 well separated from b. If p is the number of terms in the expansion, the error

behaves as

$$error < K\frac{(p+c)(c+1)+c^2}{c(c-1)}c^{-p-1}, \tag{2.15}$$

where K is a constant independent of c and p. This bound is valid for $p \geq \max\{2, \frac{2r_{b_1}c}{r_{b_1}c-r_b}\}$, where $c = \frac{|z_b - z_{b_1}|-r_b}{r_{b_1}}$.

- T_3 shifts the center of a Taylor expansion within the region of analyticity. No error is introduced by this translation operator.

2.2.3. Informal description of the algorithm. First of all, the closed unit disk is recursively subdivided into a hierarchy of meshes. Each cell at level l that contains more than s prevertices is subdivided into four child cells which are added to the level $l+1$, and then the same procedure is applied to the level $l+1$. When there are no more childless cells at a certain level L that contain more than s prevertices, the recursion terminates.

In the next stage, for each nonempty childless cell b, a multipole expansion Φ_b is constructed using Theorem 2.3. Starting from level L and going up to level 1, for every parent cell B the multipole expansion Φ_B is constructed. This is not done directly using Theorem 2.3, which would have cost $O(N)$, but by using the translation operator T_1 to shift the centers of the multipole expansions of its children to the center of B and then merging these expansions to obtain Φ_B, which has cost $O(p^2)$. At this stage of the algorithm we can evaluate the integrand at a point z inside the unit disk by including the effect of distant prevertices (i.e., prevertices outside $T(U_b) \cup T(W_b)$) by evaluating the appropriate multipole expansions and by including the effect of nearby prevertices directly. Still, we want to do more. We want to have associated with every childless cell a single local expansion that gives the effect of all the distant particles on the cell. This is made possible by the second part of the algorithm.

By definition, each cell in V_b is well separated from b, so the translation operator T_2 can be applied to convert the multipole expansion of each cell in V_b into a local expansion about the center of b and to merge them together to obtain Γ_b. We also want to include the effect of prevertices in X_b, but since cells in X_b are not well separated from b, we have to transform the effect of each single prevertex in $T(X_b)$ into a local expansion about the center of b and merge these expansions to obtain Δ_b. This is done for all cells b.

Finally, to obtain local expansions due to distant prevertices for each cell b at level 1, we define $\Psi_b = \Gamma_b + \Delta_b$. Then, using the translation operator T_3 for every parent cell B at level l, starting from level 1 and going down to level $L-1$, the expansion Ψ_B is shifted to each child cell b and added to Γ_b and Δ_b to obtain Ψ_b. With this the algorithm is complete.

Let us see how the resulting expansions can be used to evaluate the integrand at a point z inside the closed unit disk. First we find the childless cell b containing z. The sum (2.2) can then be evaluated at z by evaluating the expansion Ψ_b at z, which gives the effect of distant prevertices, and directly adding to the result the effect of each prevertex in $T(U_b) \cup T(W_b)$.

2.2.4. Geometry of the computational cell. Since the error bounds in Theorem 2.3 and the error bounds for translation operators T_1 and T_2 depend on the geometry of cells involved, we need to consider the geometry more closely, especially since the geometry of a cell changes depending on the level of refinement. Also, we need to justify our choice of the first step of the refinement. We shall look at this first step more generally.

TABLE 1
Optimum value of parameter a giving largest c for a fixed k. Based on such data we made the choice k = 8 for the initial subdivision.

k	a	c
2	0.9999	0.9099
3	0.9999	1.2925
4	0.9999	1.6110
5	0.9086	1.7577
6	0.8011	1.8228
7	0.7030	1.8497
8	0.6258	1.8694
9	0.5636	1.8844
10	0.5125	1.8962
11	0.4698	1.9056
12	0.4336	1.9134
13	0.4026	1.9200
14	0.3756	1.9255
15	0.3521	1.9302

In the first level of refinement we subdivide the unit disk into $k+1$ child cells $b_0, b_1, \ldots, b_k$, where $b_0 = \{z : |z| \le 1-a\}$ and $b_j = \{re^{i\theta} : 1-a < r \le 1,\ 2\pi(j-1)/k < \theta \le 2\pi j/k\}$ for $j = 1, 2, \ldots, k$, where $0 < a < 1$. Let us, for the moment, fix the numbers k and a and consider how this choice affects the error bounds.

We wish to find a value c that can be used in both Theorem 2.3 and the error estimate for the translation operator T_2 (2.15). Since child cells differ in size and shape and are not just scaled copies of their parents, the value of c will depend on the level of the hierarchy. This value does not vary much with the level, however, so we just take the minimum over all the levels to obtain a universally applicable lower bound for the constant c.

The larger the value of c is, the better the asymptotic error bound is in Theorem 2.3 and (2.15). Hence for a fixed k we now vary a to find the optimal value giving the largest lower bound for c. Data obtained in such a way is displayed in Table 1.

This leaves us with the algorithmic question of what is the best choice of k, the number of child cells along the circle into which the disk is subdivided at the first step. We reason as follows. The only requirement on the constant c is that it be greater than 1. Hence we can choose any $k \ge 3$. The question is, which is the optimal choice? Since we choose p so that $p \sim -\log_c(\epsilon)$ and the running time is proportional to p, it is natural to choose k so as to maximize c. By increasing k, a is decreasing; i.e., the cell b_0 is increasing in size, and cells b_j, $j = 1, \ldots, k$, are decreasing. The latter means that there are fewer levels in the hierarchy, but the former has a bad effect, since evaluation of the sum (2.2) at a point $z \in b_0$ has cost of order $O(k(s+p))$; hence as we increase k, these evaluations become more costly and the subset of the unit disk, in which this kind of an evaluation is necessary, grows. Hence we need to find a compromise, and thus choose k appropriately. As stated earlier, we have chosen $k = 8$.

2.2.5. Algorithm analysis. In [4] the FMM is used to evaluate the potential due to N charges at the positions of these N charges, whereas in our application we need to be able to evaluate the SC integrand at any point inside the unit disk. Still, the same argument shows that the cost of constructing all the expansions is $O(N)$. The number of levels in the mesh hierarchy grows as $O(\log(N/s))$. To evaluate the integrand at a single point z, first the childless cell containing it needs to be found,

which takes $O(\log(N/s))$ time, and then the expansions need to be evaluated at z and the effect of nearby prevertices taken into account directly, which takes $O(p+s)$ time. Hence the time to evaluate the function at a single point is $O(\log N)$.

3. Parameter problem. If the positions of the prevertices are known a priori, then FMM is excellent for the application to conformal mapping since once all the expansions are formed the map can be evaluated very quickly. If we want to solve the parameter problem iteratively, then at every iteration a new refinement and new expansions need to be formed.[1] The cost of doing this is $O(N)$ at each iteration. Since we want to deal with systems where N is large, the nonlinear solver used to solve the parameter problem should minimize the number of times the refinement needs to be recreated and avoid costly operations in terms of N. In particular, inverting an $N \times N$ matrix at each iteration would be unacceptable.

Driscoll and Trefethen [9, 35] formulate the problem as an unconstrained system of nonlinear equations that can be solved by various standard optimization algorithms such as quasi-Newton methods. As they carry this out, however, the work is $O(N^3)$. If one pursued this approach, one would have to turn to large-scale iterations that avoid dense linear algebra.

Another method has been suggested by Davis [7], an iterative method based on geometrical assumptions on the problem. As is well known, if the prevertices are chosen incorrectly, the resulting polygon will have the correct turning angles at the vertices but incorrect side lengths. Davis suggests that in the case of the map from the half-plane, the prevertices should simply be rescaled according to the scalings indicated by the errors in the side lengths. He implements this idea for symmetric polygons and reports very good results. Similar methods have been investigated by a number of authors [6, 14], including some earlier than Davis (see [2]).

This method has the property that it does not require any expensive calculations at each iteration; the work needed to find a new guess from the old one is $O(N \log N)$. But it is not obvious whether this method converges in general and whether many iterations are needed for high accuracy when it does converge. We have found that in practice, Davis's method is usually excellent, though not infallible. We illustrate the capabilities of this method when applied to the problem of mapping the unit disk to the inside of polygons approximating the Koch snowflake fractal.

3.1. Davis's method and the Koch snowflake. First we define the polygons approximating the fractal. Let P_1 be an equilateral triangle. Construct polygon P_{m+1} by dividing each side of P_m into three equal parts, adding a vertex where these parts meet and adding another vertex in the exterior of P_m such that the three new vertices form an equilateral triangle of side length a third of the original side (see Figure 7).

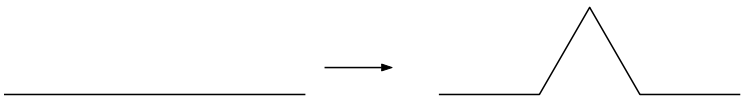


FIG. 7. *Step performed on each edge of P_m to construct the Koch snowflake polygon P_{m+1}.*

Both the fractal and the polygonal approximations, if centered at the origin and rotated as in Figure 8, are symmetric in radial lines at angles that are integer multiples of $\pi/6$. Hence, to find a map from the unit disk to the inside of the polygon, we need

[1] One might attempt some speed-ups here, but we have not done so.

only to find the positions of prevertices with argument between 0 and $\pi/6$; the others can be obtained by reflection. Suppose that there are $M+1$ prevertices with argument in $[0, \pi/6]$. We set up an iteration involving the arguments of the prevertices $\theta_j^{(n)}$, where we fix $\theta_0^{(n)} = 0$ and require $\theta_j^{(n)} \le \theta_{j+1}^{(n)}$ for all $n \ge 1$. Let w_i be the vertices of the target polygon and let $w_i^{(n)}$ be the corresponding vertices of the polygon obtained as the image of the unit disk at the nth iteration. At each iteration, the constants A and C in (2.1) are kept fixed (usually $A = 0$ and $C = 1$); at the end of the iteration they can be adjusted to rotate, translate, and scale the image of the unit disk to match the target polygon. Thus we set up the following Davis iteration:

$$\theta_{j+1}^{(n+1)} := \theta_j^{(n+1)} + k\left(\theta_{j+1}^{(n)} - \theta_j^{(n)}\right)\frac{|w_{j+1} - w_j|}{|w_{j+1}^{(n)} - w_j^{(n)}|}, \quad j = 0, \ldots, M-1, \tag{3.1}$$

where k is chosen so that $\theta_M^{(n+1)} = \pi/6$. A more stable version of the algorithm is obtained if the iteration is done on the differences of prevertices $\phi_j^{(n)} = \theta_{j+1}^{(n)} - \theta_j^{(n)}$ instead:

$$\phi_j^{(n+1)} := k\phi_j^{(n)}\frac{|w_{j+1} - w_j|}{|w_{j+1}^{(n)} - w_j^{(n)}|}, \quad j = 0, \ldots, M-1. \tag{3.2}$$

The method is easy to implement and gives good results. We have been able to solve the parameter problem of size 16,384 to a satisfactory accuracy for the polygon P_9 with 196,608 vertices in 2.3 hours on a Pentium III 800MHz processor; see Figures 8–10. More detailed results of experiments can be seen in Table 2 and Figures 11–12. In Figure 12 we can see that the time appears to be growing linearly with the number of vertices. Similar results were obtained for all the symmetric polygons we have tried. Some of these will be described in the next section.

We are not the first to carry out numerical computations on the Koch snowflake. For example, Lapidus et al. calculated beautiful images of eigenmodes of Koch snowflake drums via finite difference methods [26]; these results have been realized as mathematical sculptures by the artist Helaman Ferguson [12].

3.2. Davis's method for general polygons. In the previous example only one solution of the parameter problem exists. When a polygon has no symmetry or just a single symmetry, it is not obvious how best to force the map to be unique. One possibility is to fix three prevertices, say $\theta_0^{(n)} = \theta_0$, $\theta_1^{(n)} = \theta_1$, and $\theta_{N-1}^{(n)} = \theta_{N-1}$, and set up iteration (3.2) for $j = 1, \ldots, N-2$, choosing k so that $\theta_{N-1}^{(n+1)} = \sum_{j=0}^{N-2} \phi_j^{(n)}$ is equal to θ_{N-1}.

Suppose that the iteration converges to some values ϕ_j^*. Assuming that these values are nonzero, we get

$$|w_{j+1}^* - w_j^*| = k|w_{j+1} - w_j|, \quad j = 1, \ldots, N-2.$$

If this relation could be proved for cases $j = 0$ and $j = N-1$, with $w_N = w_0$, this would mean that w_j^* are vertices of a polygon that is a scaled, rotated, and translated copy of the target polygon. Choosing appropriate constants A and C, we see that the solution of the parameter problem would also have been found. Indeed, these two cases are implied by a more general result; see Lemma 3.1.

Unfortunately, in practice the above method performs poorly. The most reliable method we have found is that of fixing three prevertices θ_{i_0}, θ_{i_1}, and θ_{i_2} and performing three partial Davis iterations on the prevertices that lie between the three fixed ones. If

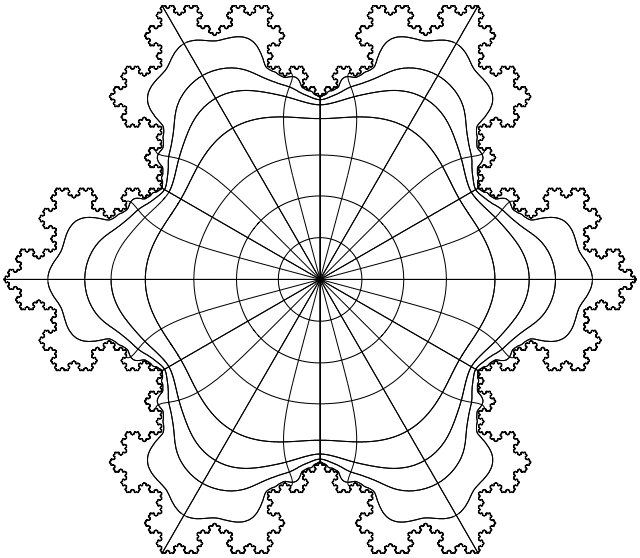


FIG. 8. *Conformal map of the unit disk to the Koch snowflake polygon* P_9 *with* 196,608 *vertices. The curves are the images of* 24 *equally spaced radii in the unit disk and of concentric circles of radii* $0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 0.99$. *The map is accurate to* 5 *or more digits everywhere.*

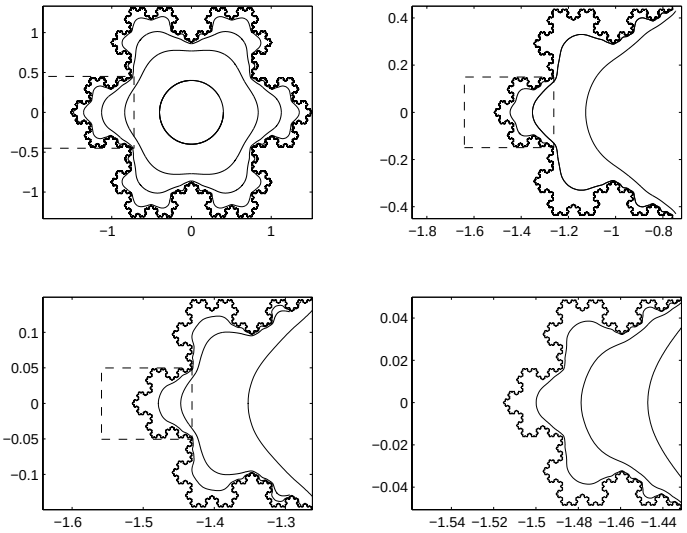


FIG. 9. *Repetition of Figure* 8 *with map shown at increasingly finer scales. Each successive plot is of the area indicated by the dashed box in the previous plot. The curves in the first plot are the images of concentric circles of radii* $0.4, 0.8, 0.95$, *and* 0.995. *In each of the three successive plots, an image of one more concentric circle is added; the radii are* $0.9995, 0.9999$, *and* 0.99999.

the iteration converges to some ϕ_j^* and these are nonzero, then, assuming $i_0 = 0$, we have

$$|w_{j+1}^* - w_j^*| = k_0|w_{j+1} - w_j|, \quad j = 0, \ldots, i_1 - 1,$$
$$|w_{j+1}^* - w_j^*| = k_1|w_{j+1} - w_j|, \quad j = i_1, \ldots, i_2 - 1,$$
$$|w_{j+1}^* - w_j^*| = k_2|w_{j+1} - w_j|, \quad j = i_2, \ldots, N - 1,$$

TABLE 2

Results for Koch snowflake polygons. The columns represent the number of vertices, time spent during 30 *iterations constructing the adaptive mesh and the expansions, total time spent for* 30 *iterations, and maximum error of the map after* 30 *iterations.*

N	Mesh time(s)	Time(s)	Max error
192	2.2	7.2	$2.6e-07$
768	11.8	28.5	$3.1e-07$
3072	54.0	116.0	$4.1e-07$
12288	224.0	486.5	$6.1e-07$
49152	927.0	1982.3	$8.6e-07$
196608	3844.2	8248.2	$5.2e-07$

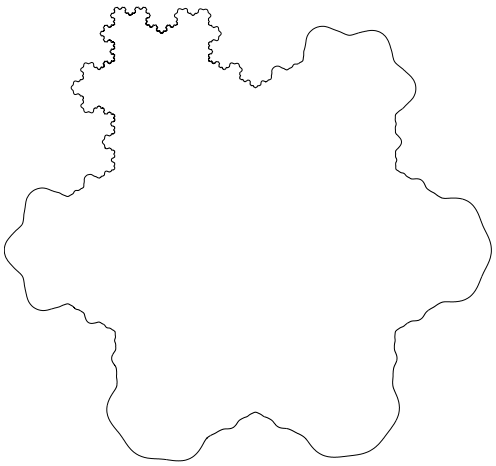


FIG. 10. *Image of a shifted circle, centered at a point* z_0 *with* $|z_0| = 0.01$ *and radius* 0.99*, under the conformal map of the unit disk to the Koch snowflake polygon* P_9*.*

where $w_N = w_0$ and $w_N^* = w_0^*$. The following result shows that $k_0 = k_1 = k_2$ and hence that a solution of the parameter problem has been obtained.

LEMMA 3.1. *Let* $P \in \mathbb{C}$ *be a polygon with vertices* $v_0, v_1, \ldots, v_{N-1}$*, and let* Q *be a polygon with vertices* $w_0, w_1, \ldots, w_{N-1}$ *such that the angles at corresponding vertices* v_i *and* w_i *are the same. If for some* $k_0, k_1, k_2 \in (0, \infty)$ *and some* i_1 *and* i_2 *such that* $0 < i_1 < i_2 < N$ *and* v_0*,* v_{i_1}*, and* v_{i_2} *are not collinear,*

$$
\begin{aligned}
|w_{j+1} - w_j| &= k_0 |v_{j+1} - v_j|, \quad j = 0, \ldots, i_1 - 1, \\
|w_{j+1} - w_j| &= k_1 |v_{j+1} - v_j|, \quad j = i_1, \ldots, i_2 - 1, \\
|w_{j+1} - w_j| &= k_2 |v_{j+1} - v_j|, \quad j = i_2, \ldots, N - 1,
\end{aligned} \tag{3.3}
$$

then $k_0 = k_1 = k_2$.

Proof. Because v_0, v_{i_1}, and v_{i_2} are not collinear, these three points are the vertices of a triangle. By the assumption of equal angles and the assumption (3.3), the three points w_0, w_{i_1}, and w_{i_2} form a triangle that is similar to the first one. The result now amounts to the statement that if two triangles are similar, then their side lengths are related by a fixed ratio. ∎

In practice we would often choose $i_0 = 0$, $i_1 = \lfloor N/3 \rfloor$, $i_2 = \lfloor 2N/3 \rfloor$. In most cases this choice would suffice, but in some cases different values have to be used to guarantee convergence. We have not found a polygon for which some choice of i_0, i_1, and i_2 does not give convergence, unless heavy crowding occurs.

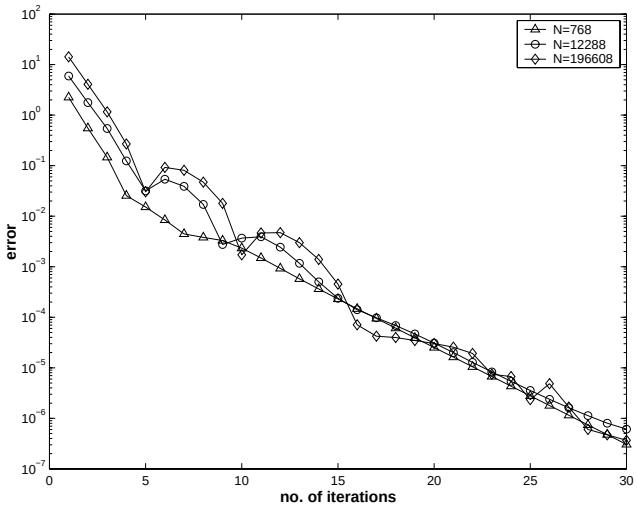


FIG. 11. *Convergence of Davis's iteration for Koch snowflake polygons* P_5, P_7, P_9.

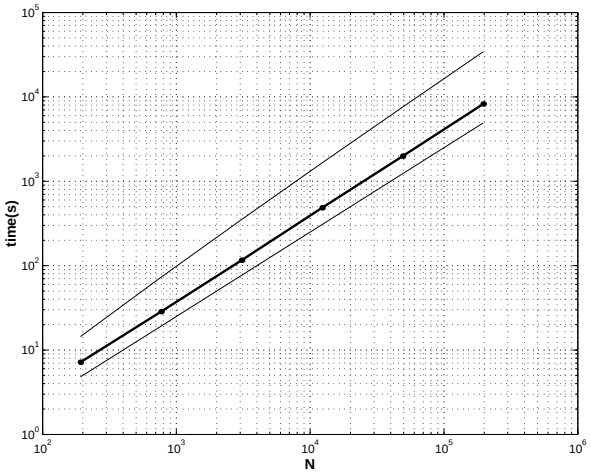


FIG. 12. *Time needed to compute* 30 *iterations for Koch snowflake polygons. Lines* $O(N \log N)$ *and* $O(N)$ *are drawn for comparison.*

As an example we consider the problem of computing the conformal modulus (resistance) of a circular L-shaped region; see Figure 13. This problem is equivalent to the problem of determining the ideal flow past an infinite square array of cylinders, and it was introduced by Rayleigh [33] and further studied by a number of authors [3, 24, 28].

By approximating the circular section of the boundary with a large number of polygonal sides, and then by mapping the region conformally to a rectangle, we can find the conformal modulus, which is the ratio of the sizes of two consecutive sides of the rectangle. This was done using the SC FMM and Davis's algorithm to find the prevertex arrangement for a map from the unit disk to the L-shaped region and then using the SC toolbox to find the appropriate rectangle and hence the resistance.

The single symmetry of the polygon was not used. The above algorithm was used

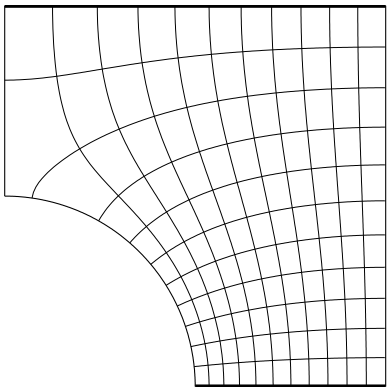


FIG. 13. *Conformal map of a circular L-shaped region onto a rectangle. The circular arc is approximated by a large number of straight line segments.*

with the choices $i_0 = 0$, $i_1 = 2$, and $i_2 = 4$, where the vertices are numbered in counterclockwise order with the bottom left corner being the initial vertex. Vertices w_0, w_1, w_2, and w_3 are mapped to the corners of the rectangle. For values of M larger than 60 we were able to obtain a good initial guess from the previous solution; with this guess Davis's iteration converged rapidly. For the data obtained using this method see Table 3 and Figure 14. This problem is also well suited for the application of Symm's integral equation [19]. This method has been implemented by Hough in his Fortran package CONFPACK [20]. Using the double precision version of this package, the number is 1.4889206978953, which is estimated to be correct in all but perhaps the final digit [21]. This agrees to all 13 digits with a number computed years ago by Moler by expanding in a series the solution of an equivalent Laplace problem [28]. Our solution matches this solution to 9 digits. In 1964, Keller and Sachs computed this number using finite differences for the same problem and obtained the figure 1.489 [24]. Bjørstad and Grosse solved this problem by means of their conformal mapping program for circular polygons and obtained the figure 1.48892070 [3].

We do not recommend SC mappings with huge numbers of sides as a competitive technique for regions like this with such simple curved boundaries; rather, its virtue is its great flexibility in being applicable to more complicated regions.

TABLE 3

Convergence of the conformal modulus of the circular L-shaped region to 1.488920697895 . . . *as the number of discretization points increases.*

M	Discretization error	Modulus
60	$8.6e-05$	1.4888513003
120	$2.1e-05$	1.4889033153
240	$5.4e-06$	1.4889163472
480	$1.3e-06$	1.4889196088
960	$3.3e-07$	1.4889204025
1920	$8.4e-08$	1.4889206296
3840	$2.1e-08$	1.4889206802
7680	$5.2e-09$	1.4889206929

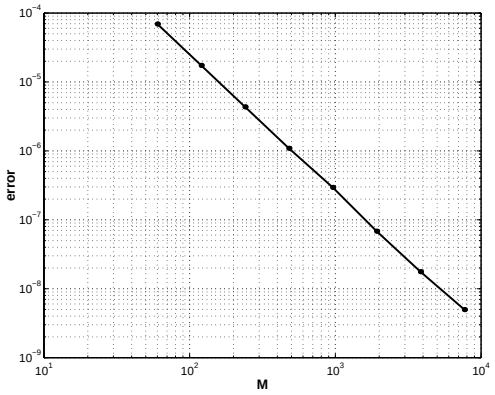


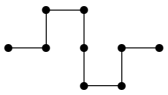
FIG. 14. *Convergence of the conformal modulus of the circular L-shaped region as the number of sides M approximating the circular boundary is increased.*

4. Further examples.

4.1. Koch curves. Koch curves are fractals that can be defined recursively—generalizations of the Koch snowflake; see [27]. The initial polygon, the *initiator*, is often taken to be an equilateral triangle or a square. At each step of the recursion, each side of the current polygon is replaced by a curve called the *generator*. We have already used the equilateral triangle as the initiator for the Koch snowflake. Since both these initiators have rotational symmetries, the fractals have the same rotational symmetries. Also it can be easily shown that the prevertices of an SC conformal map that sends 0 to the center of the rotational symmetry also have these symmetries. This can be used to reduce the size of the parameter problem.

We have used Davis's algorithm to find the map f such that $f(0) = 0$. All fractals are centred at 0, and we fix one prevertex to make the map unique.

Fractal A. The initiator is the unit square, and the generator is the following:



$$\text{Hausdorff dimension} = \frac{3}{2}.$$

A plot of the map and the convergence can be seen in Figures 15 and 16. The polygon has 16,384 vertices, and the time needed for the parameter problem was 1.8 hours. As can be seen in Figure 16 the convergence stops after reaching a relatively low accuracy. The reason for this is that at that point of the solution, the closest two prevertices are 6.7×10^{-15} apart.

Fractal B. Again the initiator is the square, and the generator is

$$\text{Hausdorff dimension} = \frac{\log 4}{\log 3}.$$

A plot of the map and the convergence can be seen in Figures 17 and 18. The polygon has 4096 vertices, and the time needed for the parameter problem was 12.5 minutes.

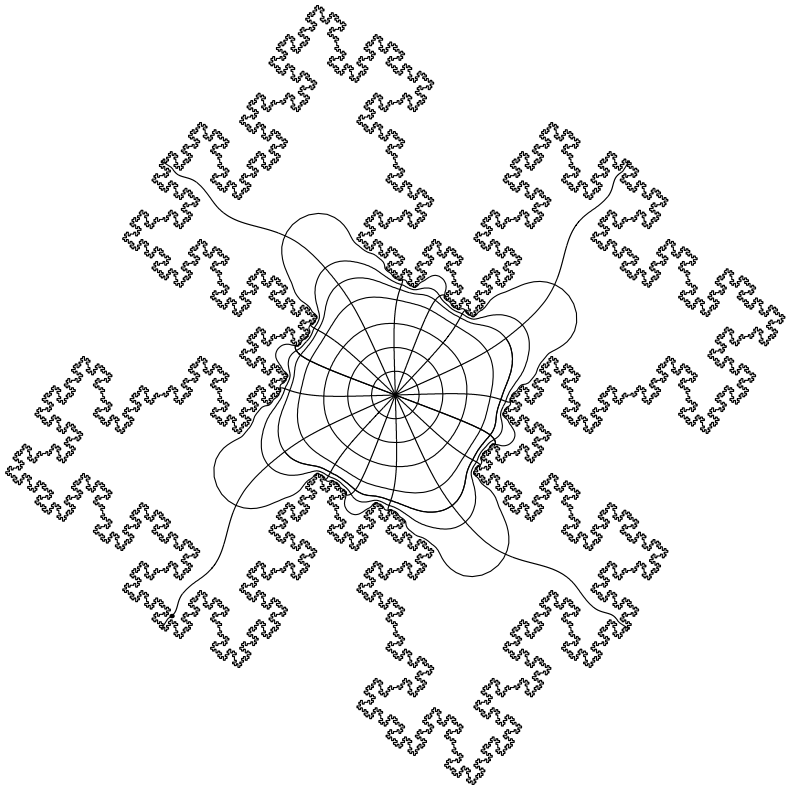


FIG. 15. *Conformal map of the unit disk to the fractal A (see section* 4.1*). The curves are the images of* 16 *equally spaced radii in the unit disk and of concentric circles of radii* $0.2, 0.4, 0.6, 0.8, 0.9, 0.95,$ *and* 0.99.

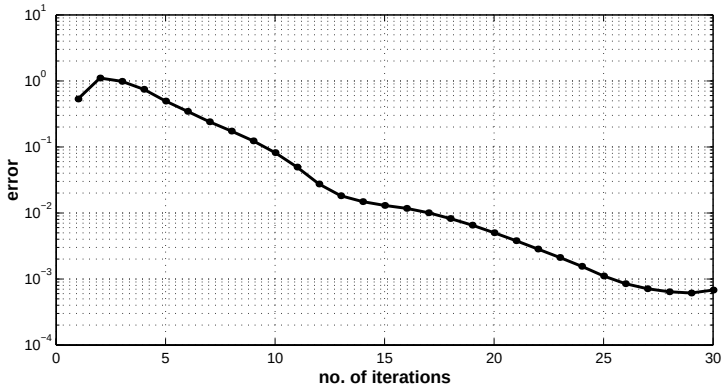


FIG. 16. *Convergence for fractal A.*

4.1.1. Regions with smooth sides. Let us first consider an example of a region consisting of joined circular arcs (Figure 19). We use the symmetry of the region, and Davis's algorithm converges rapidly and cleanly. See Figure 19 for the map and Figure 20 for a plot of the convergence. Such convergence is often seen when only a few corners exist and the rest of the boundary is smooth. For the plots we

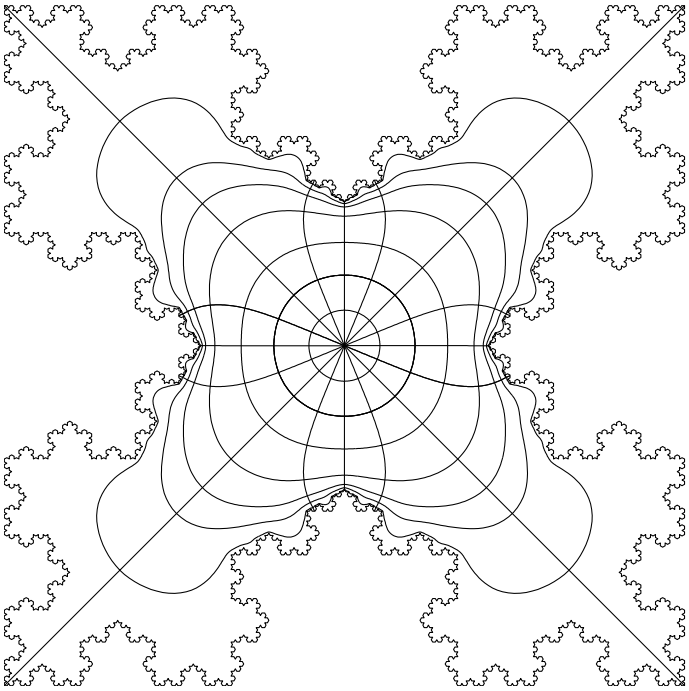


FIG. 17. *Conformal map of the unit disk to fractal B (see section* 4.1*). The curves are the images of* 16 *equally spaced radii in the unit disk and of concentric circles of radii* $0.2, 0.4, 0.6, 0.8, 0.9, 0.95,$ *and* 0.99.

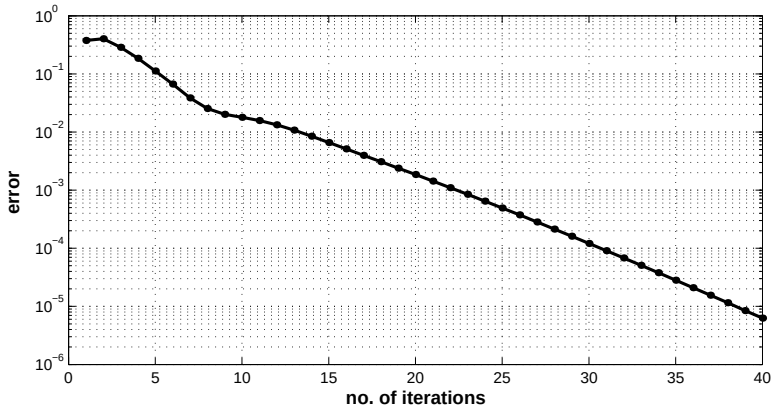


FIG. 18. *Convergence for fractal B.*

used 250 sides to approximate each smooth arc (discretization error 3.5×10^{-6}).

If we add six slits to the region the situation changes. The convergence is not as fast as before and, due to crowding, it stops once some of the prevertices are too close to each other. See Figure 21 for the map and Figure 22 for a plot of the convergence. These results were obtained by approximating each smooth arc by 30 sides (more sides result in too much crowding). At the last iteration the closest two prevertices were 5×10^{-10} apart.

Since we have noticed that smooth regions tend to behave well for Davis's

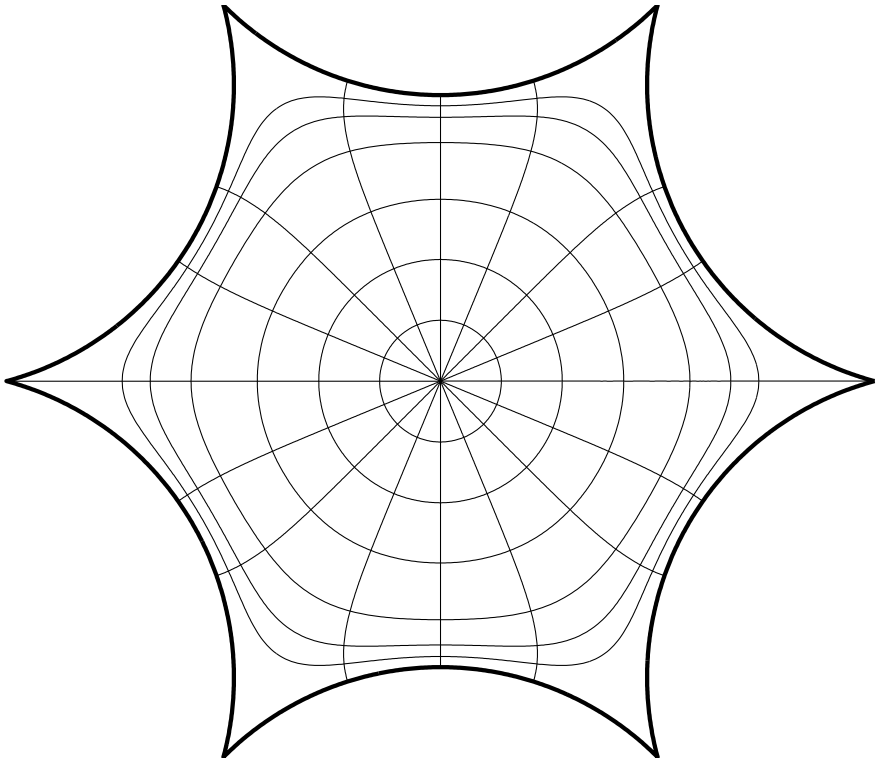


FIG. 19. *A domain with piecewise smooth boundary has been approximated by a polygon with* 250 *sides approximating each curved subarc. The curves are the images of* 16 *equally spaced radii in the unit disk and of concentric circles of radii* $0.2, 0.4, 0.6, 0.8, 0.9,$ *and* 0.95.

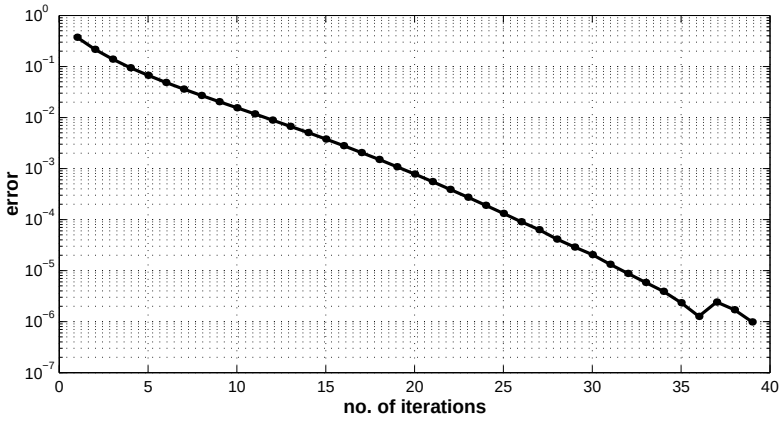


FIG. 20. *Convergence for the piecewise smooth region in Figure* 19.

algorithm, let us consider a completely smooth region, an inverted ellipse, with the boundary slightly perturbed, then observe how the convergence behaves as these perturbations reduce to zero.

To obtain the vertices of the perturbed inverted ellipse polygon we evaluate the

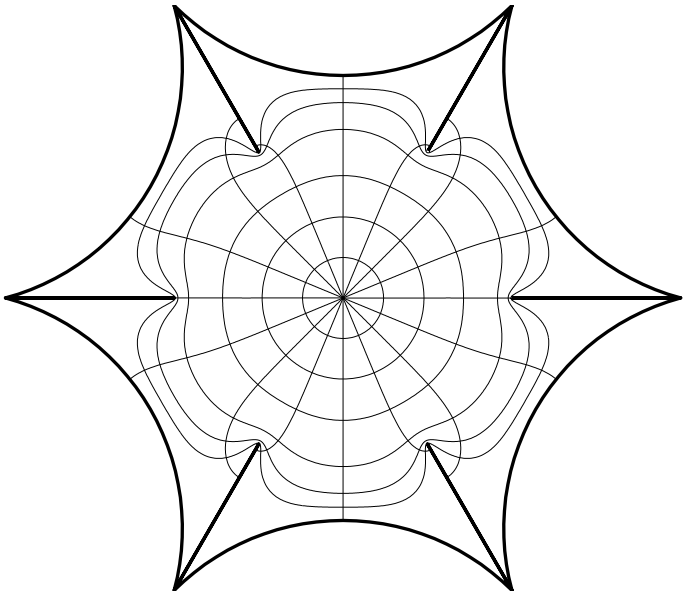


FIG. 21. *Conformal map from the unit disk to the region from Figure* 19 *but with added slits. The curves are the images of* 16 *equally spaced radii in the unit disk and of concentric circles of radii* $0.2, 0.4, 0.6, 0.8, 0.9,$ *and* 0.95.

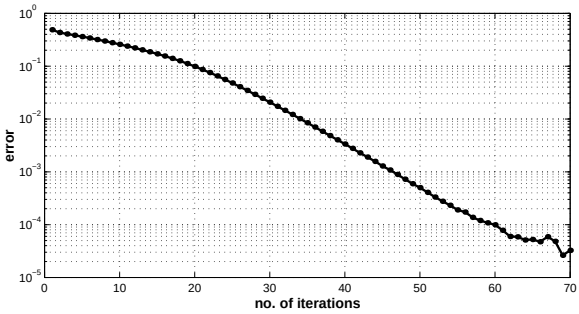


FIG. 22. *Slower convergence due to the additional slits in the region.*

following formula at points $t_k = 2\pi(k-1)/N$:

$$f(t) = (0.1 + h\rho(t))\sqrt{1-(1-q^2)\cos(t)^2}e^{it}, \quad t \in [0, 2\pi), \tag{4.1}$$

where ρ is a function that returns a random number between 0 and 1, q determines the shape of the smooth inverted ellipse, and h is the parameter that controls how much the boundary is allowed to oscillate from the smooth shape. For an example of both a perturbed polygon ($h = 0.025$) and a smooth one see Figure 23, and for the convergence see Figure 24. In Table 4 we show the number of iterations needed to obtain error $< 10^{-5}$ for various values of s and with $p = 0.2$.

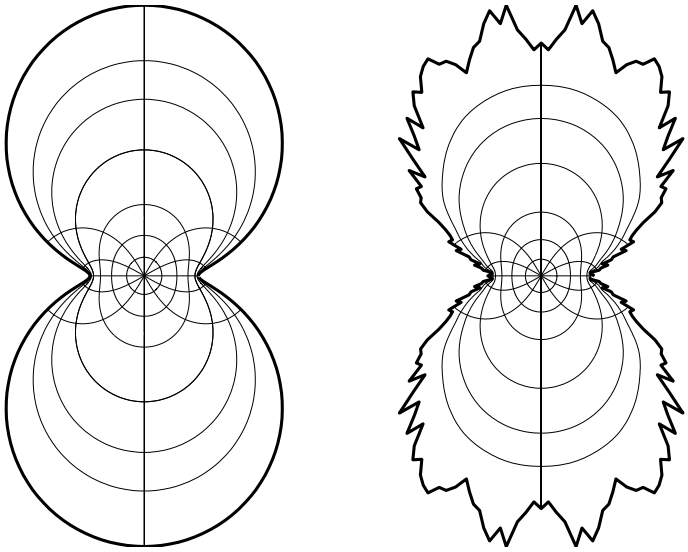


FIG. 23. *Conformal map from the unit disk to a polygonal approximation to a smooth inverted ellipse and to a distorted inverted ellipse (see section* 4.1*). The curves are the images of* 12 *equally spaced radii in the unit disk and of concentric circles of radii* $0.2, 0.4, 0.6, 0.8, 0.9,$ *and* 0.95

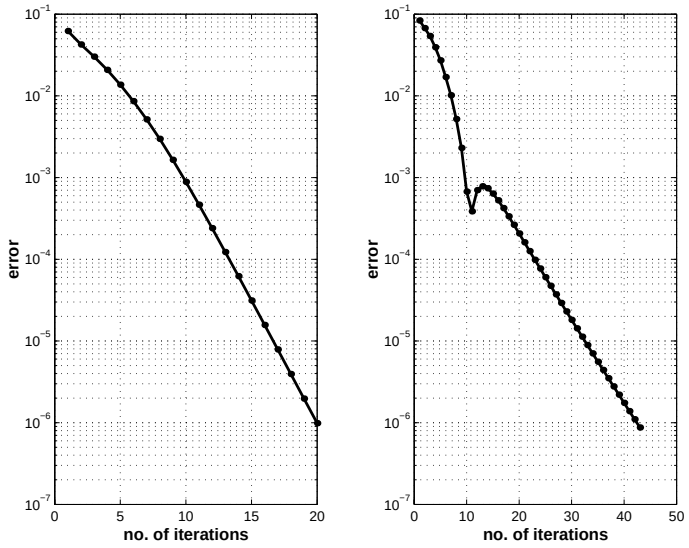


FIG. 24. *Convergence curves for the same two regions. Note the different horizontal scales.*

5. Conclusion and future work. A robust method has been presented for the computation of SC maps of polygons with many vertices. Mapping polygons with tens of thousands of vertices is now realistic. The method is also applicable to regions whose boundary, or part of the boundary, is smooth. Still, for most smooth polygons the use of other methods like Symm's integral equation is preferable.

A number of improvements to the algorithm are possible. The cost of applying a translation operator to a multipole or Taylor expansion is $O(p^2)$ in the current implementation, but methods exist that can reduce this to $O(p \log p)$ [17, 1, 11].

TABLE 4

The first column shows the degree of distortion of the inverted ellipse, and the second column shows the number of iterations needed to obtain error $< 10^{-5}$ *for the distorted polygon.*

h	No. of iterations
0.025	43
0.020	40
0.015	28
0.010	23
0.005	22
0	20

Unfortunately these methods are unstable for large values of p, and since our application requires high accuracy, this may prove to be a problem. A new representation of potentials in two dimensions, in which most translation operators are diagonal, was introduced by Hrycak and Rokhlin [22, 23] and later extended with great success to three dimensions, where the improvement is most noticeable [18, 5]. This new method does not suffer from stability problems. Reconstructing the mesh and expansions from scratch at each iteration seems wasteful especially in the later stages of convergence. Also, during the parameter solution the map need not be evaluated inside the unit disk; all the integration can be done on the boundary so that in this case a simpler fast multipole algorithm could be used. The results of these improvements will be reported at a later stage.

An important problem that has not been addressed in this paper is that of crowding. This is an intrinsic problem of conformal mapping in general. This phenomenon occurs when the domain has long, narrow channels; when considering polygons with hundreds of thousands of vertices, these channels need not be too pronounced to force two prevertices to be so close that many significant digits are lost when computing the differences between their arguments. One solution to the problem of crowding for SC mapping has been put forward by Driscoll and Vavasis [10]. Their "CRDT algorithm" uses cross-ratios and Delaunay triangulation of the polygon, which is computed in $O(N^2)$ steps. Thus this method would have to be modified before it could be used for polygons with many vertices. Otherwise a new idea is needed to combat crowding. Certainly this is an issue that needs to be addressed in the future.

Finally, we note that in principle it should be possible to extend our algorithm to handle (in the sense of lazy evaluation) fractals with infinitely many sides.

REFERENCES

[1] C. L. BERMAN, *Grid-multipole calculations*, SIAM J. Sci. Comput., 16 (1995), pp. 1082–1091.
[2] K. J. BINNS AND P. J. LAWRENSON, *Analysis and Computation of Electric and Magnetic Field Problems*, Pergamon Press, New York, 1963.
[3] P. BJØRSTAD AND E. GROSSE, *Conformal mapping of circular arc polygons*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 19–32.
[4] J. CARRIER, L. GREENGARD, AND V. ROKHLIN, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 669–686.
[5] H. CHENG, L. GREENGARD, AND V. ROKHLIN, *A fast adaptive multipole algorithm in three dimensions*, J. Comput. Phys., 155 (1999), pp. 468–498.
[6] E. COSTAMAGNA, *On the numerical inversion of the Schwarz–Christoffel conformal transformation*, IEEE Trans. Microwave Theory Tech., MTT-35 (1987), pp. 35-40.
[7] R. T. DAVIS, *Numerical methods for coordinate generation based on Schwarz–Christoffel transformations*, in Proceedings of the 4th AIAA Computational Fluid Dynamics Conference, Williamsburg, VA, 1979, pp. 1–15.

[8] T. A. Driscoll, *A MATLAB toolbox for Schwarz–Christoffel mapping*, ACM Trans. Math. Software, 22 (1996), pp. 168–186.
[9] T. A. Driscoll and L. N. Trefethen, *Schwarz–Christoffel Mapping*, Cambridge University Press, Cambridge, UK, 2002.
[10] T. A. Driscoll and S. A. Vavasis, *Numerical conformal mapping using cross-ratios and Delaunay triangulation*, SIAM J. Sci. Comput., 19 (1998), pp. 1783–1803.
[11] W. D. Elliot and J. A. Board, Jr., *Fast Fourier transform accelerated fast multipole algorithm*, SIAM J. Sci. Comput., 17 (1996), pp. 398–415.
[12] H. Ferguson, *Mathematics in Stone and Bronze*, Meridian Creative Group, Erie, PA, 1994.
[13] J. M. Floryan and C. Zemach, *Schwarz–Christoffel mappings: A general approach*, J. Comput. Phys., 72 (1987), pp. 347–371.
[14] J. M. Floryan and C. Zemach, *Schwarz–Christoffel methods for conformal mapping of regions with a periodic boundary*, J. Comput. Appl. Math., 46 (1993), pp. 77–102.
[15] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, MA, 1987.
[16] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
[17] L. Greengard and V. Rokhlin, *On the Efficient Implementation of the Fast Multipole Algorithm*, Research Report YALEU/DCS/RR-602, Yale University, New Haven, CT, 1988.
[18] L. Greengard and V. Rokhlin, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta Numer., 6 (1997), pp. 229–269.
[19] P. Henrici, *Applied and Computational Complex Analysis, Volume* 3: *Discrete Fourier Analysis, Cauchy Integrals, Construction of Conformal Maps, Univalent Functions*, Wiley, New York, 1986.
[20] D. M. Hough, *User's Guide to CONFPACK*, ETH Zürich IPS Research Report 90-11, Zürich, Switzerland, 1990.
[21] D. M. Hough, *private communication*, 2002.
[22] T. Hrycak and V. Rokhlin, *An Improved Fast Multipole Algorithm for Potential Fields*, Research Report 1089, Department of Computer Science, Yale University, New Haven, CT, 1995.
[23] T. Hrycak and V. Rokhlin, *An improved fast multipole algorithm for potential fields*, SIAM J. Sci. Comput., 19 (1998), pp. 1804–1826.
[24] H. B. Keller and D. Sachs, *Calculations of the conductivity of a medium containing cylindrical inclusions*, J. Appl. Phys., 35 (1964), pp. 537–538.
[25] N. Kerzman and M. R. Trummer, *Numerical conformal mapping via the Szegö kernel*, J. Comput. Appl. Math., 14 (1986), pp. 111–123.
[26] M. L. Lapidus, J. W. Neuberger, R. J. Renka, and C. A. Griffith, *Snowflake harmonics and computer graphics: Numerical computation of spectra on fractal drums*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 6 (1996), pp. 1185–1210.
[27] B. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman, New York, 1982.
[28] C. B. Moler, *private communication to L. N. Trefethen*, 1983.
[29] S. T. O'Donnell and V. Rokhlin, *A fast algorithm for the numerical evaluation of conformal mappings*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 475–487.
[30] A. M. Odlyzko, *The* 10^{20}*th Zero of the Riemann Zeta Function and* 175 *Million of Its Neighbors*, manuscript, 1992.
[31] A. M. Odlyzko, *The* 10^{22}*nd zero of the Riemann zeta function*, Contemp. Math., 290 (2001), pp. 139–144.
[32] A. M. Odlyzko and A. Schönhage, *Fast algorithms for multiple evaluations of the Riemann zeta function*, Trans. Amer. Math. Soc., 309 (1988), pp. 797–809.
[33] Lord Rayleigh, *On the influence of obstacles arranged in rectangular order upon the properties of a medium*, Philosophical Magazine, 34 (1892), pp. 481–502.
[34] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.
[35] L. N. Trefethen, *Numerical computation of the Schwarz–Christoffel transformation*, SIAM J. Sci. Stat. Comput., 1 (1980), pp. 82–102.
[36] L. N. Trefethen, *SCPACK User's Guide*, MIT Numerical Analysis Report 89-2, MIT, Cambridge, MA, 1989.
[37] M. R. Trummer, *An efficient implementation of a conformal mapping method based on the Szegö kernel*, SIAM J. Numer. Anal., 23 (1986), pp. 853–872.