

# A Multipole Method for Schwarz-Christoffel Mapping of Polygons with Thousands of Sides

Lehel Banjai and Lloyd 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  $N \approx 2 \times 10^5$ .

**Keywords:** conformal mapping, fast multipole method, Schwarz-Christoffel mapping, Koch snowflake

**Mathematics Subject Classification:** 30C30, 65E05, 70F10.

## 1 Introduction

Conformal mapping has been recognised for a long time 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  $\Omega$ . 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 semi-explicit formula has

been known for many years, the Schwarz-Christoffel 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 [26, 27]. The Schwarz-Christoffel 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 [5]. Detailed descriptions of algorithms implemented in these packages and of many applications of Schwarz-Christoffel mapping can be found in the recent book [6]. 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 behaviour for most polygons, not guaranteed behaviour for all polygons). These improvements are due to the use of the Fast Multipole Method (FMM) developed by Greengard and Rokhlin [3, 11, 25] and the use of a simple iteration for finding the unknown parameters in the SC formula due to Davis [4].

This is not the first time that the FMM has been applied to numerical conformal mapping. O'Donnell and Rokhlin [20] modified an algorithm [16, 28] for computing conformal maps of smooth domains via the Szegő 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  $O(N)$ .

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 [21, 22, 23], 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 Schwarz-Christoffel mapping

In this paper we only consider conformal maps from the unit disk to the interior of a polygon; exterior maps can be treated by similar methods. The Schwarz-Christoffel formula for the interior map of the unit disk to a polygon is

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

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 only affects 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

$$G(z) = \sum_{k=1}^L \beta_{k+l} \log(z - z_{k+l}), \quad (2)$$

where  $\beta_{k+l} = \alpha_{k+l} - 1$ ,  $L \leq N$  and  $0 \leq l \leq N - L$ . To simplify notation we will assume that  $l = 0$ .

First, we state and prove a theorem which allows us to apply the Fast Multipole Method to the problem at hand. Once we have this result we can show how the FMM can be used to evaluate (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 centred 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*

$$\log(z - z_1) = \log(z - z_0) - \sum_{k=1}^{\infty} \frac{1}{k} \left( \frac{z_1 - z_0}{z - z_0} \right)^k. \quad (3)$$

**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  $|w| < 1$ .

□

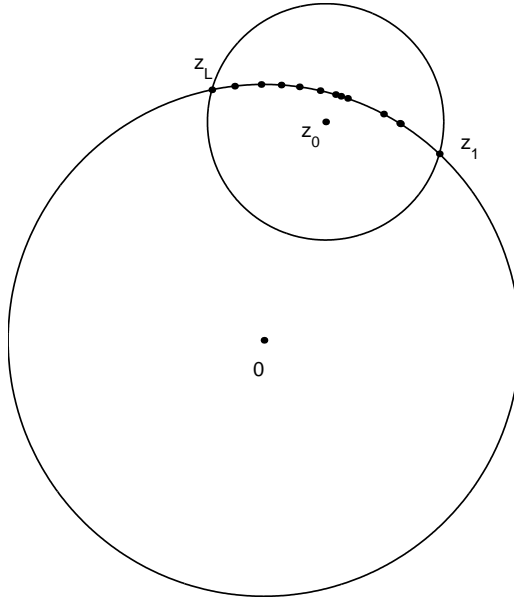


Figure 1: The set-up for Lemma 2.2.

**Lemma 2.2** *Let  $\beta_i \in [-1, 1]$ ,  $i = 1, \dots, L$ , 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  $z_1, z_2, \dots, z_L$  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  $k \geq 1$ ,*

$$\left| \sum_{i=1}^L \beta_i (z_i - z_0)^k \right| \leq (1 + k\pi)Cr^k. \quad (4)$$

**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  $\nu_i$  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

$$\sum_{j=1}^{L-1} |\nu_{j+1} - \nu_j| \leq r\pi. \quad (5)$$

Define sequences  $\omega$ ,  $b$  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  $i = 1, \dots, L$ , with boundary cases  $b_0 = w_1 = 0$ .

Two facts to notice are that

$$\|b\|_\infty \leq C \quad (6)$$

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 (5) and the fact  $|\nu_j| \leq r$

$$\begin{aligned} \|w\|_1 &\leq kr^{k-1} \sum_{j=1}^{L-1} |\nu_{j+1} - \nu_j| \\ &\leq kr^k \pi. \end{aligned} \quad (7)$$

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

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

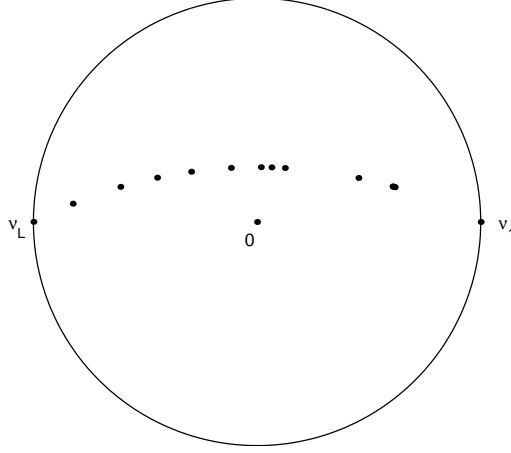


Figure 2: Circle centred at  $(0,0)$  with radius  $r$ . Points  $\omega_i$  are shown for  $k = 1$ .

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

$$\sum_{i=1}^L \beta_i \omega_i = b_L \omega_L - \sum_{i=1}^L b_{i-1} w_i. \quad (8)$$

Using (6), (7), the above expression and the fact  $|\omega_L| = r^k$ , by Hölder's inequality we obtain the required bound

$$\left| \sum_{i=1}^L \beta_i (z_i - z_0)^k \right| \leq \|b\|_\infty (r^k + \|w\|_1) = (1 + k\pi) C r^k. \quad (9)$$

□

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 Schwarz-Christoffel integrand. Also it provides an estimate of the error when this expansion is approximated by a truncated expansion of  $p$  terms.

**Theorem 2.1** *Let  $z_i, \beta_i, r, i = 1, \dots, L$ , and  $z_0$  be as in Lemma 2.2. Then  $\forall z \in \mathbb{C}$  such that  $|z - z_0| > r$ , the function  $G(z) = \log(\prod_{i=1}^L (z - z_i)^{\beta_i})$  can be expressed as*

$$G(z) = Q \log(z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k} \quad (10)$$

where

$$Q = \sum_{i=1}^L \beta_i \text{ and } a_k = - \sum_{i=1}^L \beta_i \frac{(z_i - z_0)^k}{k}. \quad (11)$$

Furthermore, for any  $p \geq 1$ ,

$$\left| G(z) - Q \log(z - z_0) - \sum_{k=1}^p \frac{a_k}{(z - z_0)^k} \right| \leq \left( \frac{A}{c - 1} \right) c^{-p} \quad (12)$$

where

$$A = (1 + \pi)C \text{ and } c = \frac{|z - z_0|}{r}. \quad (13)$$

**Proof:** The form of (10) is obtained from Lemma 2.1 and the fact

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

To obtain the error bound (12) we proceed as follows:

$$\begin{aligned} \left| G(z) - Q \log(z - z_0) - \sum_{k=1}^p \frac{a_k}{(z - z_0)^k} \right| &= \left| \sum_{k=p+1}^{\infty} \frac{a_k}{(z - z_0)^k} \right| \\ &= \left| \sum_{k=p+1}^{\infty} \sum_{i=1}^L \frac{\beta_i}{k} \left( \frac{z_i - z_0}{z - z_0} \right)^k \right| \\ &\leq \sum_{k=p+1}^{\infty} \frac{1}{k |z - z_0|^k} \left| \sum_{i=1}^L \beta_i (z_i - z_0)^k \right|. \end{aligned} \quad (14)$$

Then by Lemma 2.2 we conclude that

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

This result ensures that the fast multipole method can be applied for efficient evaluation of Schwarz-Christoffel maps with a large number of prevertices.

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

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

This error bound is fine if  $L$  is not much larger than  $p$ ; see (14). For example, if  $p = L + 1$ , the truncation error in Theorem 2.1 would be bounded by a simple expression:

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

For  $L \gg p$ , however, results of Theorem 2.1 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  $\beta_i$  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 future we intend to consider generalisations 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 2.1 and we want to evaluate the integrand at  $M$  points  $y_1, \dots, y_M$  such that  $|y_i - z_0| > r$ . To do this directly would take  $O(LM)$  operations, but to form the multipole expansion and then to evaluate it  $M$  times takes  $O(Lp + Mp)$  operations, which is a big saving 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 box and use the adaptive version of the FMM. Still, using a non-standard



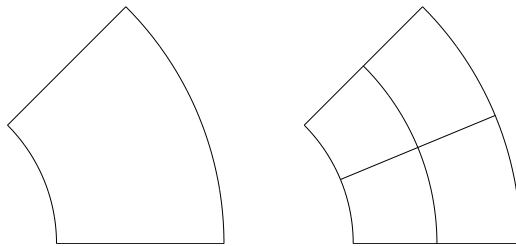


Figure 3: Subdivision of a parent box.

shape of the computational box and choosing an appropriate division results in a simpler and faster code.

We develop an adaptive procedure for the subdivision of the computational box into boxes of different shapes. We define levels of refinement. At level zero we only have one box, and at level 1 we subdivide the computational box into 9 boxes:  $b_0, b_1, \dots, b_8$  where  $b_0 = \{z : |z| \leq 0.5097\}$  and  $b_k = \{re^{i\theta} : 0.5097 < r \leq 1, 2\pi(k-1)/8 < \theta \leq 2\pi k/8\}$  for  $k = 1, 2, \dots, 8$ . The reason for choosing this particular subdivision will be explained later. Going from level  $l$  to level  $l+1$ , only boxes containing more than some fixed number  $s$  of prevertices will be subdivided. Hence the circular box  $b_0$  never gets subdivided.

A general non-empty box is bounded by two straight lines whose continuations intersect at the origin, and two circular arcs centred at the origin of which one is at distance 1 from the origin. A box like this is subdivided into four boxes by a straight line connecting the midpoints of the circular arcs and a circular arc centred at the origin and connecting the midpoints of the straight sides (see Figure 3).

We say that two boxes at the same level of refinement are *well separated* if they are not adjacent. For a box  $b$  we define the *centre*  $z_b$  to be the point equally spaced from all the corners of the box and the *radius*  $r_b$  to be the distance from the centre to the corners.

## Notation

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

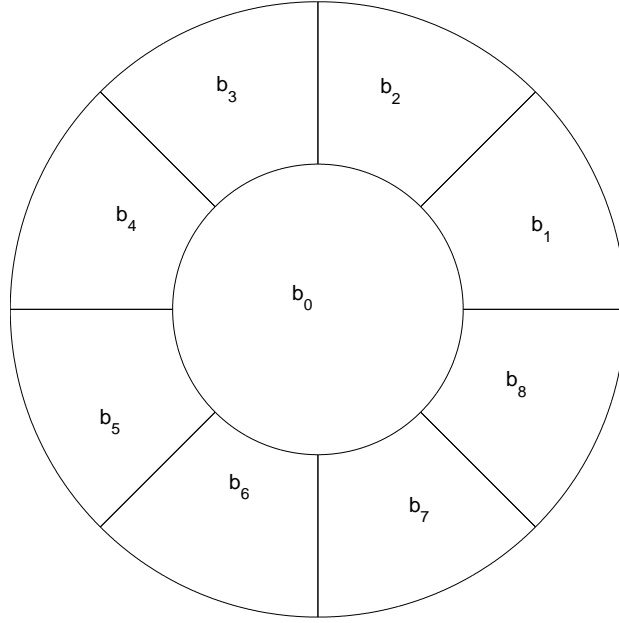


Figure 4: Boxes  $b_0, \dots, b_8$  at level 1.

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

With each box  $b$  we associate 4 different lists of boxes:

1.  $U_b$  contains  $b$  and all childless boxes adjacent to  $b$ .
2.  $V_b$  contains all the children of  $b$ 's parent that are well separated from  $b$  (note this can only happen at level 1) and the children of the colleagues of  $b$ 's parent 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.

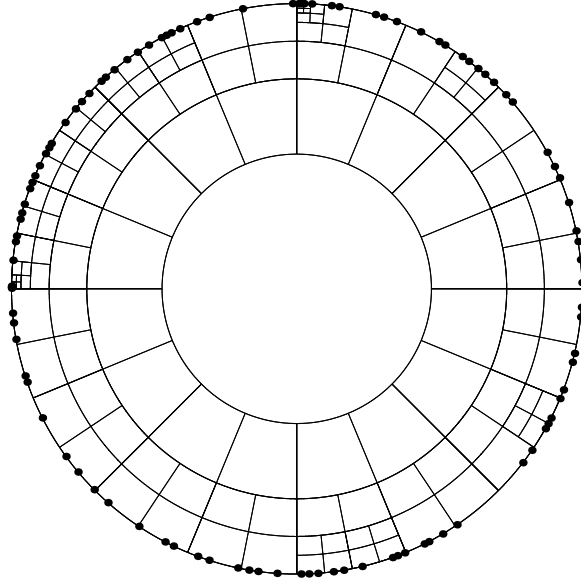


Figure 5: Complete adaptive mesh with  $s = 3$ .

4.  $X_b$  contains all boxes  $c$  such that  $b \in W_c$ .

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

- $\Phi_b$  denotes the  $p$ -term multipole expansion due to  $T(b)$ .
- $\Psi_b$  denotes the  $p$ -term local (Taylor) expansion, valid inside the box  $b$ , representing the effect of distant prevertices.
- $\Gamma_b$  denotes the  $p$ -term local expansion due to all the prevertices in  $T(V_b)$ .
- $\Delta_b$  denotes the  $p$ -term local expansion due to all the prevertices in  $T(X_b)$ .

## Translation operators

To complete the mathematical apparatus needed for the FMM we need to define three translation operators. We only give an informal description here, together with error bounds. The details can be found in [10].

$T_1$  is a translation operator that shifts the centre 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  $\Phi_b$  into a Taylor series valid in a box  $b_1$  well separated from  $b$ . If  $p$  is the number of terms in the expansion, an error bound for the truncated series is

$$error < \frac{A(4e(p+c)(c+1) + c^2)}{c(c-1)} c^{-p-1}. \quad (15)$$

where  $A$  is defined in Theorem 2.1. This bound is valid for  $p \geq \max\{2, \frac{2c}{c-1}\}$ , where  $c = \frac{|z_b - z_{b_1}| - r_b}{r_{b_1}}$ .

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

## Informal description of the algorithm

First of all the closed unit disk is recursively subdivided into a hierarchy of meshes. Each box at level  $l$  that contains more than  $s$  prevertices is subdivided into four child boxes 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 boxes at a certain level  $L$  that contain more than  $s$  prevertices, the recursion terminates.

In the next stage, for each nonempty childless box  $b$ , a multipole expansion  $\Phi_b$  is constructed using Theorem 2.1. Starting from level  $L$  and going up to level 1, for every parent box  $B$  the multipole expansion  $\Phi_B$  is constructed. This is not done directly using Theorem 2.1, which would have cost  $O(N)$ , but by using the translation operator  $T_1$  to shift the centres of the multipole expansions of its children to the centre of  $B$  and then merging these expansions to obtain  $\Phi_B$ , 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 box a single local expansion that gives the effect on the box of all the distant particles. This is made possible by the second part of the algorithm.

By definition, each box in  $V_b$  is well separated from  $b$ , so the translation operator  $T_2$  can be applied to convert the multipole expansion of each box in  $V_b$  into a local expansion about the centre of  $b$  and to merge them together to obtain  $\Gamma_b$ . We also want to include the effect of prevertices in  $X_b$ , but since boxes 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 centre of  $b$  and merge these expansions to obtain  $\Delta_b$ . This is done for all boxes  $b$ .

Finally, to obtain local expansions due to distant prevertices for each box  $b$  at level 1, we define  $\Psi_b = \Gamma_b + \Delta_b$ . Then, using the translation operator  $T_3$  for every parent box  $B$  at level  $l$ , starting from level 1 and going down to level  $L - 1$ , the expansion  $\Psi_B$  is shifted to each child box  $b$  and added to  $\Gamma_b$  and  $\Delta_b$  to obtain  $\Psi_b$ . 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 box  $b$  containing  $z$ . The sum (2) can then be evaluated at  $z$  by evaluating the expansion  $\Psi_b$  at  $z$ , which gives the effect of distant particles, and directly adding to the result the effect of each prevertex in  $T(U_b) \cup T(W_b)$ .

## Geometry of the computational box

Since the error bounds in Theorem 2.1 and the error bounds for translation operators  $T_1$  and  $T_2$  depend on the geometry of boxes involved, we need to consider the geometry more closely, especially since the geometry of a box 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.

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

We wish to find a value  $c$  that can be used in both Theorem 2.1 and in the error estimate for the translation operator  $T_2$  (15). Since child boxes 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 constant  $c$ .

Inspecting the error bounds in Theorem 2.1 and (15) we see that we need to maximise  $c$ . To find optimal values of  $k$  and  $a$  we fix  $k$  and then find  $a$

$k$	$a$	$c$
3	0.6056	0.7297
4	0.6101	1.0861
5	0.5927	1.2798
6	0.5674	1.4292
7	0.5389	1.5441
8	0.5097	1.6331
9	0.4812	1.7026
10	0.4543	1.7578
11	0.4292	1.8021
12	0.4005	1.8228
13	0.3745	1.8374
14	0.3515	1.8497
15	0.3311	1.8603

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

so that  $c$  can be chosen as large as possible. Data obtained in such way is displayed in Table 1.

The only requirement on the constant  $c$  is that it is greater than 1. Hence we can choose any  $k \geq 4$ . 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 maximise  $c$ . By increasing  $k$ ,  $a$  is decreasing, i.e., the box  $b_0$  is increasing in size and boxes  $b_j$ ,  $j = 1, \dots, k$  are decreasing. The latter means that there are fewer levels in the hierarchy, but the former has a bad effect. Evaluation of the sum (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 choose  $k$  appropriately. As stated earlier we have chosen  $k = 8$ .

## Algorithm analysis

In [3] 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 Schwarz-Christoffel integrand at any point inside the unit disk. Still, the same argument shows that the time needed to construct 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 box 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<sup>1</sup>. 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.

Trefethen and Driscoll [6, 26] 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 [4], 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-

---

<sup>1</sup>One might attempt some speedups here, but we have not done so.

lengths. He implements this idea for symmetric polygons and reports very good results.

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, though not infallible, Davis's method is usually excellent. 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.

## 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 6).

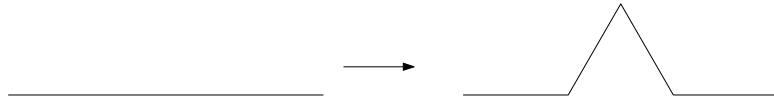


Figure 6: Step performed on each edge of  $P_m$  to construct the Koch snowflake polygon  $P_{m+1}$

Both the fractal and the polygonal approximations, if centred at the origin and rotated as in Figure 7, 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 only need 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  $\theta_j^{(n)} \leq \theta_{j+1}^{(n)}$ , for all  $n \geq 1$ . Let  $w_i$  be the vertices of the target polygon and let  $w_i^{(n)}$  be the corresponding vertices of the polygon



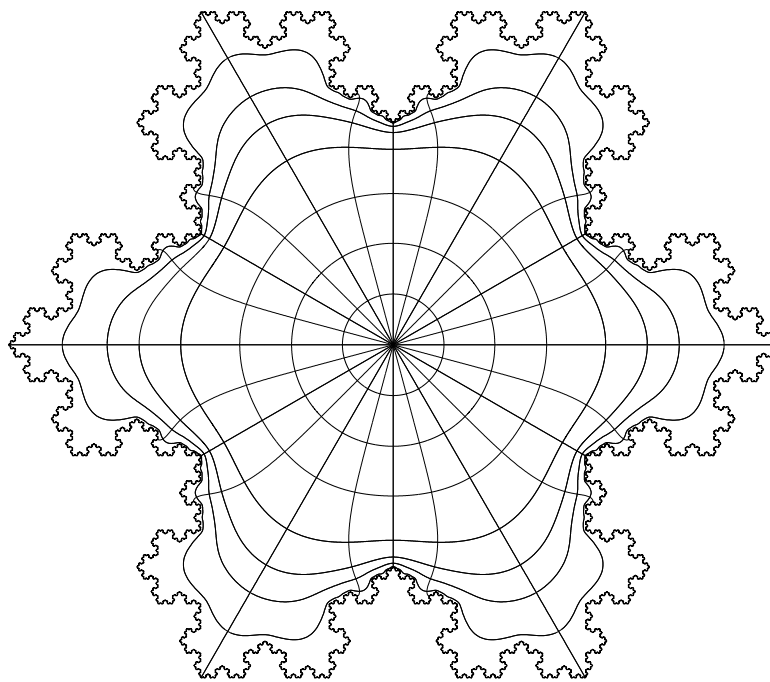


Figure 7: Conformal map of the unit disk to the Koch snowflake polygon  $P_9$  with 196608 vertices. The curves are the images of 24 equally spaced radii in the unit disk and of concentric circles of radii 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 0.99. The map is accurate to 5 or more digits everywhere.

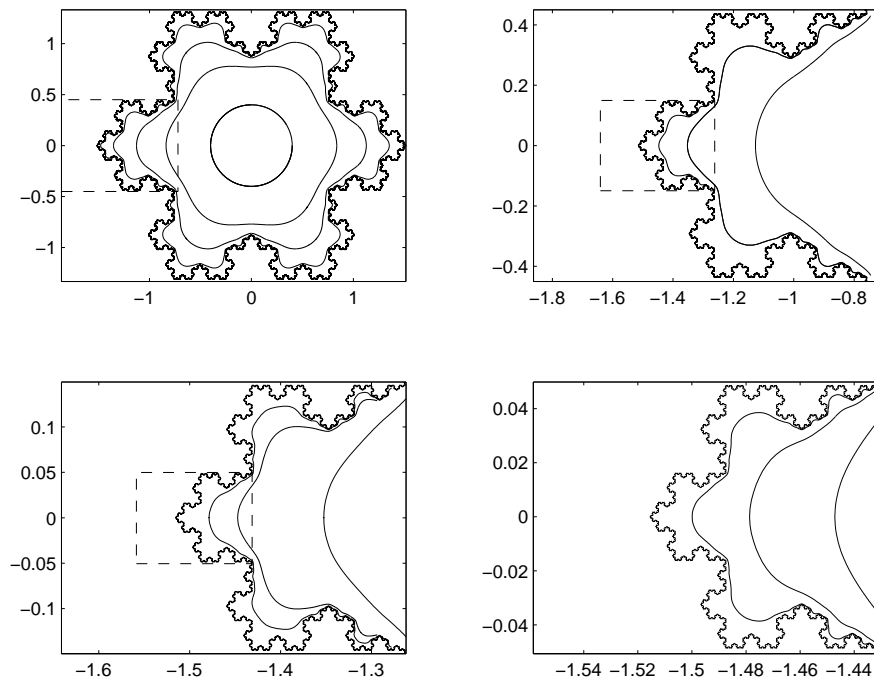


Figure 8: Repetition of Figure 7 with map shown at increasingly finer scales. Each successive plot is of the area indicated by the dashed box in the previous one. The curves in the first plot are the images of concentric circles of radii 0.4, 0.8, 0.95 and 0.995. In each of the three successive plots image of one more concentric circle is added; the radii are 0.9995, 0.9999 and 0.99999.

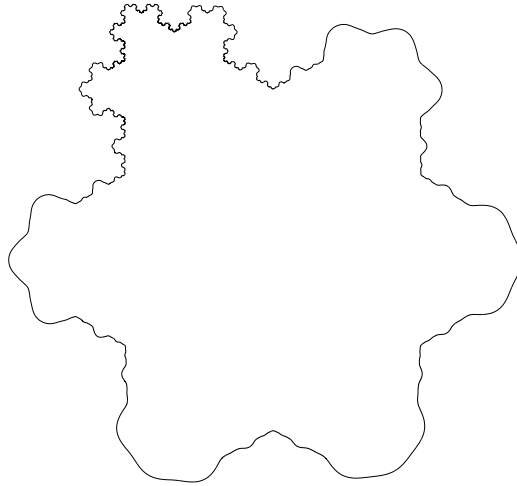


Figure 9: Image of a shifted circle, centred at a point  $z_0$  with  $|z_0| = 0.01$  and radius 0.99, under the conformal map of the unit disk to the Koch snowflake polygon  $P_9$ .

obtained as the image of the unit disk at  $n$ th iteration. At each iteration, the constants  $A$  and  $C$  in (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:

$$\theta_{j+1}^{(n+1)} := \theta_j^{(n+1)} + k \left( \theta_{j+1}^{(n)} - \theta_j^{(n)} \right) \frac{|w_{j+1} - w_j|}{|w_{j+1}^{(n)} - w_j^{(n)}|}, \quad j = 0, \dots, M-1; \quad (16)$$

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,

$$\phi_j^{(n+1)} := k \phi_j^{(n)} \frac{|w_{j+1} - w_j|}{|w_{j+1}^{(n)} - w_j^{(n)}|}, \quad j = 0, \dots, M-1. \quad (17)$$

The method is easy to implement, and gives good results. We have been able to solve the parameter problem of size 16384 to a satisfactory accuracy for the polygon  $P_9$  with 196608 vertices in 4.3 hours on a Pentium III 800MHz

$N$	mesh time(s)	time(s)	max error
192	4.1	7.5	$2.6e - 07$
768	12.1	32.0	$3.1e - 07$
3072	51.3	152.9	$4.1e - 07$
12288	201.5	757.5	$6.1e - 07$
49152	669.1	3402.2	$8.6e - 07$
196608	2060.96	15444.2	$5.2e - 07$

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.

processor; see Figures 7–9. More detailed results of experiments can be seen in Table 2 and Figures 10–11. 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 Koch snowflake. For example, Lapidus, et al. have calculated beautiful images of eigenmodes of Koch snowflake drums via finite difference methods [17], which have been realized as mathematical sculptures by the artist Helaman Ferguson [9].

## Davis’s method for general polygons

In the previous example one and 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  $\theta_{n,0} = \theta_0$ ,  $\theta_{n,1} = \theta_1$  and  $\theta_{n,N-1} = \theta_{N-1}$ , and set up iteration (17) for  $j = 1, \dots, N - 2$ , choosing  $k$  so that  $\theta_{N-1}^{(n+1)} = \sum_{j=0}^{N-2} \phi_j^{(n)}$  is equal to  $\theta_{N-1}$ .

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

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

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

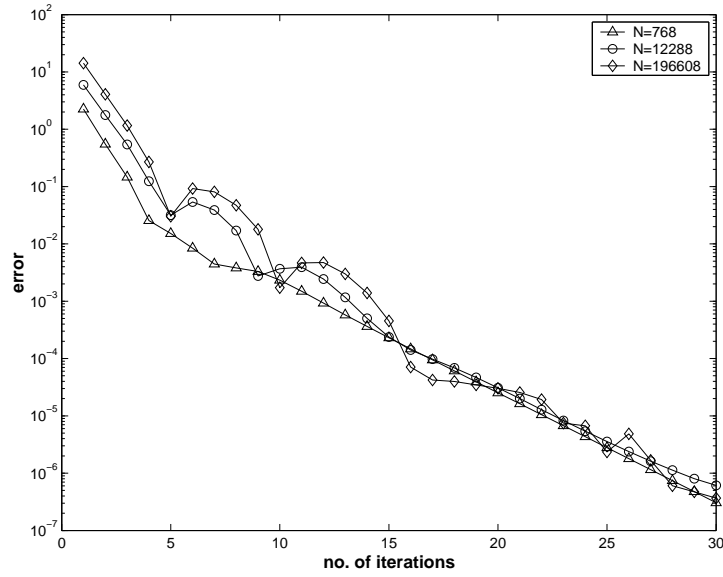


Figure 10: Convergence of Davis's iteration for Koch snowflake polygons  $P_5, P_7, P_9$ .

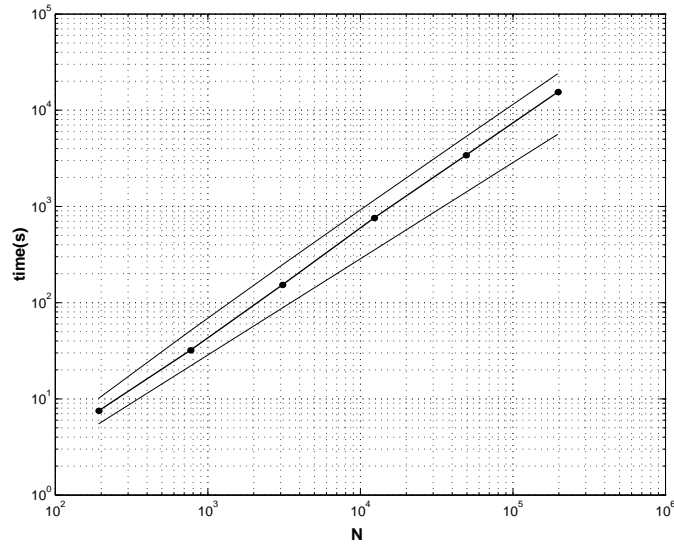


Figure 11: Time needed to compute 30 iterations for Koch snowflake polygons. Lines  $O(N \log N)$  and  $O(N)$  are drawn for comparison.

and translated copy of the target polygon. Choosing appropriate constants  $A$  and  $C$  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  $\theta_{i_0}$ ,  $\theta_{i_1}$  and  $\theta_{i_2}$ , and performing three Davis iterations on the prevertices that lie between the three fixed ones. If the iteration converges to some  $\phi_j^*$  and these are nonzero, then, assuming  $i_0 = 0$ , we have

$$\begin{aligned} |w_{j+1}^* - w_j^*| &= k_0 |w_{j+1} - w_j|, & j = 0, \dots, i_1 - 1, \\ |w_{j+1}^* - w_j^*| &= k_1 |w_{j+1} - w_j|, & j = i_1, \dots, i_2 - 1, \\ |w_{j+1}^* - w_j^*| &= k_2 |w_{j+1} - w_j|, & j = i_2, \dots, N - 1, \end{aligned}$$

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  $v_0, v_1, \dots, v_{N-1}$ , and let  $Q$  be a polygon with vertices  $w_0, w_1, \dots, w_{N-1}$  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,*

$$\begin{aligned} |w_{j+1} - w_j| &= k_0 |v_{j+1} - v_j|, & j = 0, \dots, i_1 - 1, \\ |w_{j+1} - w_j| &= k_1 |v_{j+1} - v_j|, & j = i_1, \dots, i_2 - 1, \\ |w_{j+1} - w_j| &= k_2 |v_{j+1} - v_j|, & j = i_2, \dots, N - 1, \end{aligned} \tag{18}$$

*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 (18), 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.

As an example we consider the problem of computing the conformal modulus (resistance) of a circular L-shaped region; see Figure 12. 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 [24] and further studied by a number of authors [2, 15, 19].

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 fast multipole method 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 with the choice  $i_0 = 0$ ,  $i_1 = 2$  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 13. This problem is also well suited for the application of Symm's integral equation [13]. This method has been implemented by Hough in his Fortran package CONFPACK [14]. Using this package with the best allowed accuracy of  $10^{-6}$ , the number obtained was 1.488920199, which is correct to 6 digits. In 1964 Keller and Sachs computed this number using finite differences for the same problem and obtained the figure 1.489 [15]. Bjørstad and Grosse solved this problem by means of their conformal mapping program for circular polygons and obtained the figure 1.48892070 [2]. The most accurate result was computed by Moler by expanding in a series the solution of an equivalent Laplace problem [19]. Moler claims that the number 1.488920697895 is accurate to 12 or 13 digits, which agrees with our result to 9 digits.

We do not recommend Schwarz-Christoffel 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.

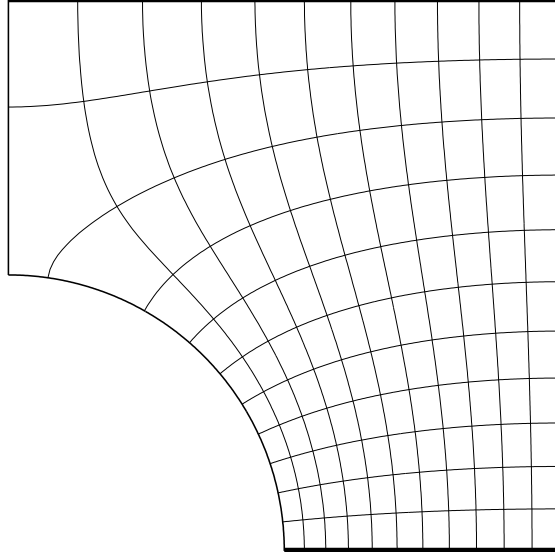


Figure 12: Conformal map of a circular L-shape region onto a rectangle. The circular arc is approximated by a large number of straight line segments.

## 4 Further examples

### 4.1 Koch curves

Koch curves are fractals that can be defined recursively, generalisations of the Koch snowflake. 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 a SC conformal map that sends 0 to the centre 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.



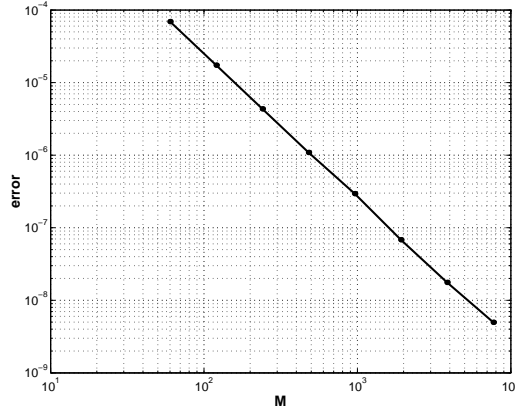


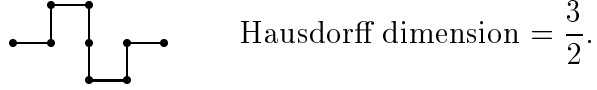
Figure 13: Convergence of the conformal modulus of the circular L-shaped region as the number of sides  $M$  approximating the circular boundary is increased.

$M$	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

Table 3: Convergence of the conformal modulus of the circular L-shaped region to  $1.488920697895 \dots$  as the number of discretization points increases.

## Fractal A

The initiator is the unit square, and the generator is the following:



A plot of the map and the convergence can be seen in Figures 14 and 15. The polygon has 16384 vertices and the time needed for the parameter problem was 1.8 hours. As can be seen in Figure 15 the convergence stops very after reaching a relatively low accuracy. The reason for this is that at that point of the solution, the closest two prevertices are  $6.7 \times 10^{-15}$  apart.

## Fractal B

Again the initiator is the square, and the generator is:



A plot of the map and the convergence can be seen in Figures 16 and 17. The polygon has 4096 vertices and the time needed for the parameter problem was 12.5 minutes.

## 4.2 Regions with smooth sides

Let us first consider an example of a region consisting of joined circular arcs (Figure 18). We use the symmetry of the region and Davis's algorithm converges rapidly and cleanly. See Figure 18 for the map, and Figure 19 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 used 250 sides to approximate each smooth arc (discretization error  $3.5 \times 10^{-6}$ ).

If we add 6 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 20 for the map, and

$s$	no. of iterations
0.025	43
0.020	40
0.015	28
0.010	23
0.005	22
0	20

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.

Figure 21 for a plot of the convergence. These results were obtained by approximating each smooth arc by 30 sides (more sides results in too much crowding). At the last iteration the closest two prevertices were  $5 \times 10^{-10}$  apart.

Since we have noticed that smooth regions tend to behave well for Davis's 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 following formula at points  $t_k = 2\pi(k-1)/N$ :

$$f(t) = (0.1 + sr(t))\sqrt{1 - (1 - p^2)\cos(t)^2}e^{it}, \quad t \in [0, 2\pi), \quad (19)$$

where  $r$  is a function that returns a random number between 0 and 1,  $p$  determines the shape of the smooth inverted ellipse and  $s$  is the parameter that controls how much is the boundary allowed to oscillate from the smooth shape. For an example of both a perturbed polygon ( $s = 0.025$ ) and a smooth one see Figure 22 and for the convergence see Figure 23. In Table 4 we show the number of iterations needed to obtain error  $< 10^{-5}$  for various values of  $s$  and with  $p = 0.2$ .

## 5 Conclusion and future work

A robust method has been presented for the computation of Schwarz-Christoffel 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  $O(p \log p)$  [12, 1, 8]. Unfortunately these methods are unstable for large values of  $p$ , and since our application requires high accuracy this may prove to be a problem. 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 some of these possible improvements will be reported at a later stage.

An important problem that has not been addressed in this paper is the problem 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 a 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 of the problem of crowding for Schwarz-Christoffel mapping has been put forward by Driscoll and Vavasis [7]. 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] L. BERMAN, *Grid-multipole calculations*, SIAM J. Sci. Comput., 16 (1995), pp. 1082–91.
- [2] P. BJØRSTAD AND E. GROSSE, *Conformal mapping of circular arc polygons*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 19–32.
- [3] J. CARRIER, L. GREENGARD AND V. ROKHLIN, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 669–86.
- [4] R. T. DAVIS, *Numerical methods for coordinate generation based on Schwarz-Christoffel transformations*, In 4th AIAA Comput. Fluid Dynamics Conf., Williamsburg, VA, 1979, pp. 1–15.
- [5] T. A. DRISCOLL, *A MATLAB toolbox for Schwarz-Christoffel mapping*, ACM Trans. Math. Soft., 22 (1996), pp. 168–86.
- [6] T. A. DRISCOLL AND L. N. TREFETHEN, *Schwarz-Christoffel Mapping*, Cambridge University Press, Cambridge UK, 2002.
- [7] T. A. DRISCOLL AND S. A. VAVASIS, *Numerical conformal mapping using cross-ratios and Delaunay triangulation*, SIAM J. Sci. Comput., 19 (1998), pp. 1783–803.
- [8] W. D. ELLIOT AND J. A. BOARD, *Fast Fourier transform accelerated fast multipole algorithm*, SIAM J. Sci. Comput., 17 (1996), pp. 398–415.
- [9] H. FERGUSON, *Mathematics in Stone and Bronze*, Meridian Creative Group, Erie, 1994.
- [10] L. GREENGARD, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, Massachusetts, 1987.
- [11] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
- [12] L. GREENGARD AND V. ROKHLIN, *On the efficient implementation of the fast multipole algorithm*, Research Report YALEU/DCS/RR-602, February 1988.

- [13] P. HENRICI, *Applied and Computational Complex Analysis, Volume 3: Discrete Fourier Analysis, Cauchy Integrals, Construction of Conformal Maps, Univalent Functions*, Wiley, New York, 1986.
- [14] D. M. HOUGH, *User's guide to CONFPACK*, ETH Zürich IPS Research Report 90-11, 1990.
- [15] H. B. KELLER AND D. SACHS, *Calculations of the conductivity of a medium containing cylindrical inclusions*, J. Appl. Phys., 35 (1964), pp. 537–38.
- [16] N. KERZMAN AND M. R. TRUMMER, *Numerical conformal mapping via the Szegő kernel*, J. Comput. Appl. Math., 14 (1986), pp. 111–23.
- [17] 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. 1185–210.
- [18] B. B. MANDELBROT, *The Fractal Geometry of Nature*, W. H. Freeman and Company, New York, 1982.
- [19] C. B. MOLER, private communication to L. N. Trefethen, 1983.
- [20] S. T. O'DONNELL AND V. ROKHLIN, *A fast algorithm for the numerical evaluation of conformal mappings*, SIAM J. Sci. Stat. Comput., 3 (1989), pp. 475–87.
- [21] A. M. ODLYZKO, *The  $10^{20}$ -th zero of the Riemann zeta function and 175 million of its neighbors*, unpublished manuscript, 1992, available at <http://www.research.att.com/~amo/unpublished/index.html>.
- [22] A. M. ODLYZKO, *The  $10^{22}$ -nd zero of the Riemann zeta function*, Contemp. Math., 290 (2001), pp. 139–44.
- [23] A. M. ODLYZKO AND A. SCHÖNHAGE, *Fast algorithms for multiple evaluations of the Riemann zeta function*, Trans. Am. Math. Soc., 309 (1988), pp. 797–809.
- [24] LORD RAYLEIGH, *On the influence of obstacles arranged in rectangular order upon the properties of a medium*, Philosophical Magazine, 34 (1892), pp. 481–502.

- [25] V. ROKHLIN, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187-207.
- [26] L. N. TREFETHEN, *Numerical computation of the Schwarz-Christoffel transformation*, SIAM J. Sci. Stat. Comput., 1 (1990), pp. 82-102.
- [27] L. N. TREFETHEN, SCPACK user's guide, MIT Numerical Analysis Report 89-2 (1989).
- [28] M. R. TRUMMER, *An efficient implementation of a conformal mapping method based on the Szegő kernel*, SIAM J. Numer. Anal., 23 (1986), pp. 853-872.

LLOYD N. TREFETHEN, OXFORD UNIVERSITY COMPUTING LABORATORY, WOLFSON BUILDING, PARKS ROAD, OXFORD OX1 3QD, UK

*E-mail address:* LNT@comlab.ox.ac.uk

LEHEL BANJAI, OXFORD UNIVERSITY COMPUTING LABORATORY, WOLFSON BUILDING, PARKS ROAD, OXFORD OX1 3QD, UK

*E-mail address:* lehelb@comlab.ox.ac.uk

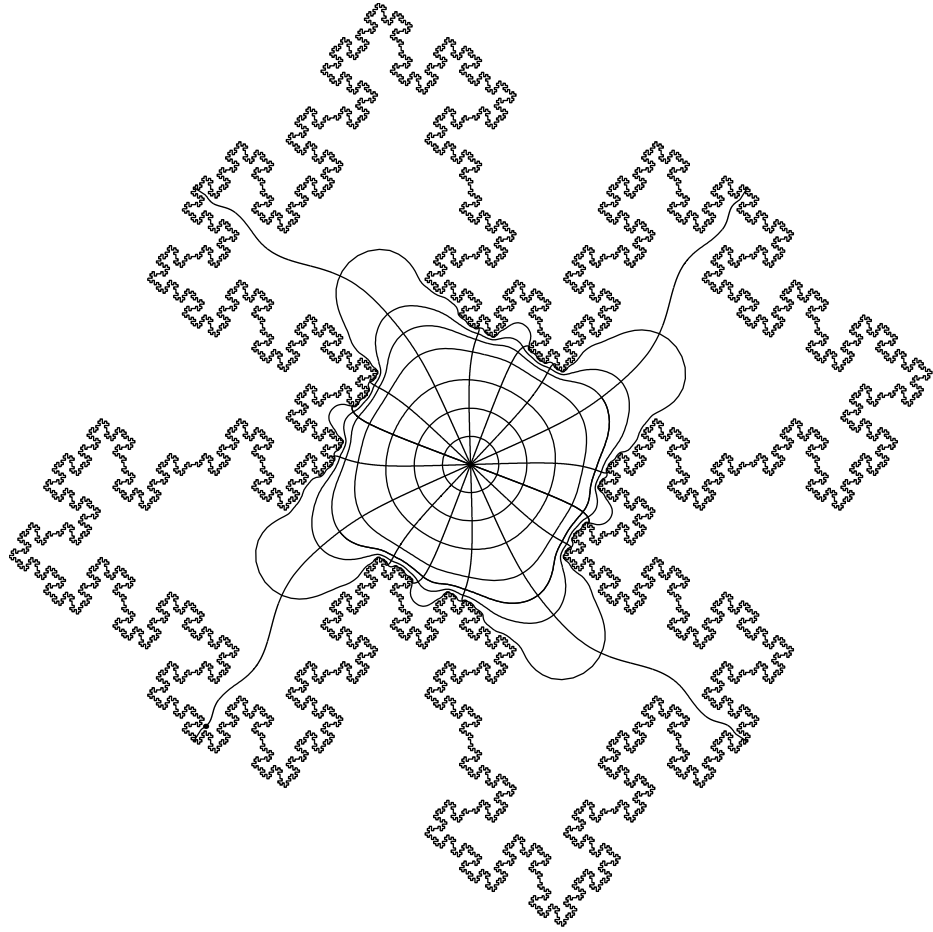


Figure 14: Conformal map of the unit disk to the fractal A (see Section 4.1). The curves are the images of 16 equally spaced radii in the unit disk and of concentric circles of radii 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, and 0.99.

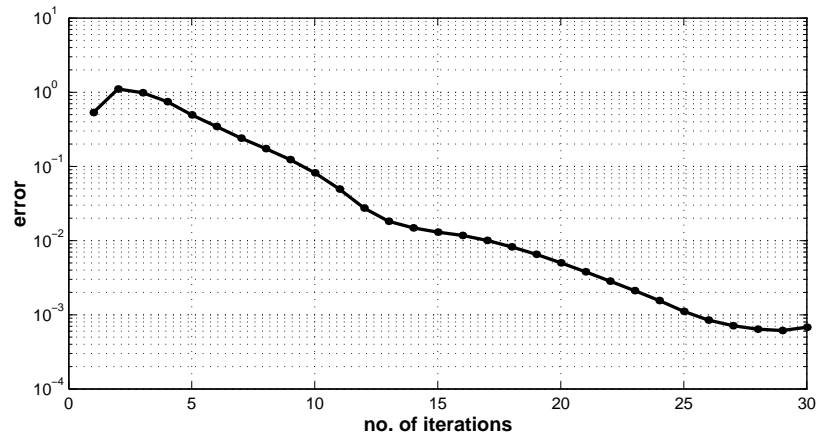


Figure 15: Convergence for fractal A



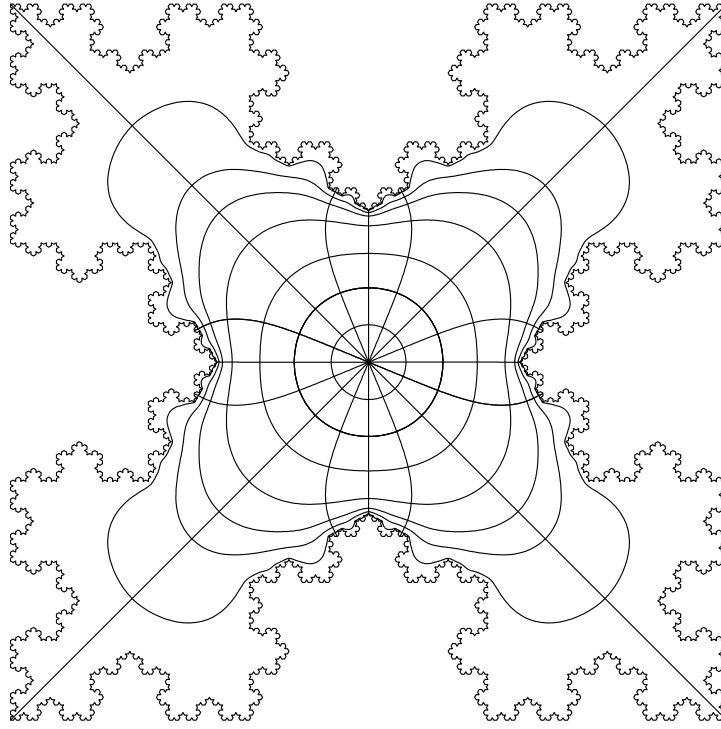


Figure 16: Conformal map of the unit disk to the fractal B (see Section 4.1). The curves are the images of 16 equally spaced radii in the unit disk and of concentric circles of radii 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, and 0.99.

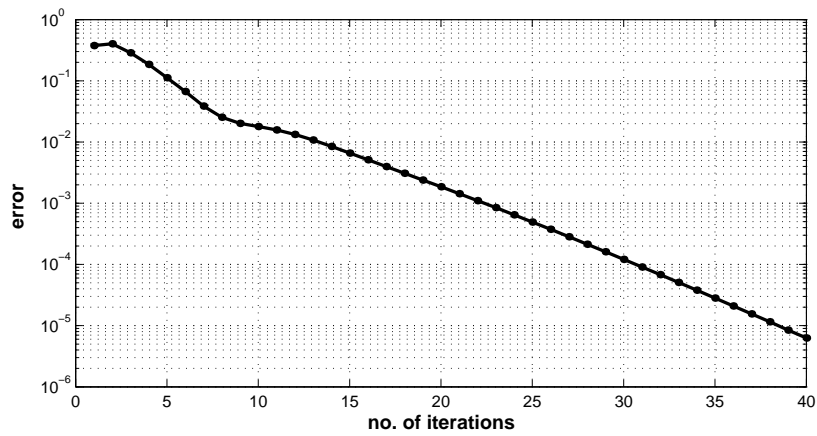


Figure 17: Convergence for fractal B

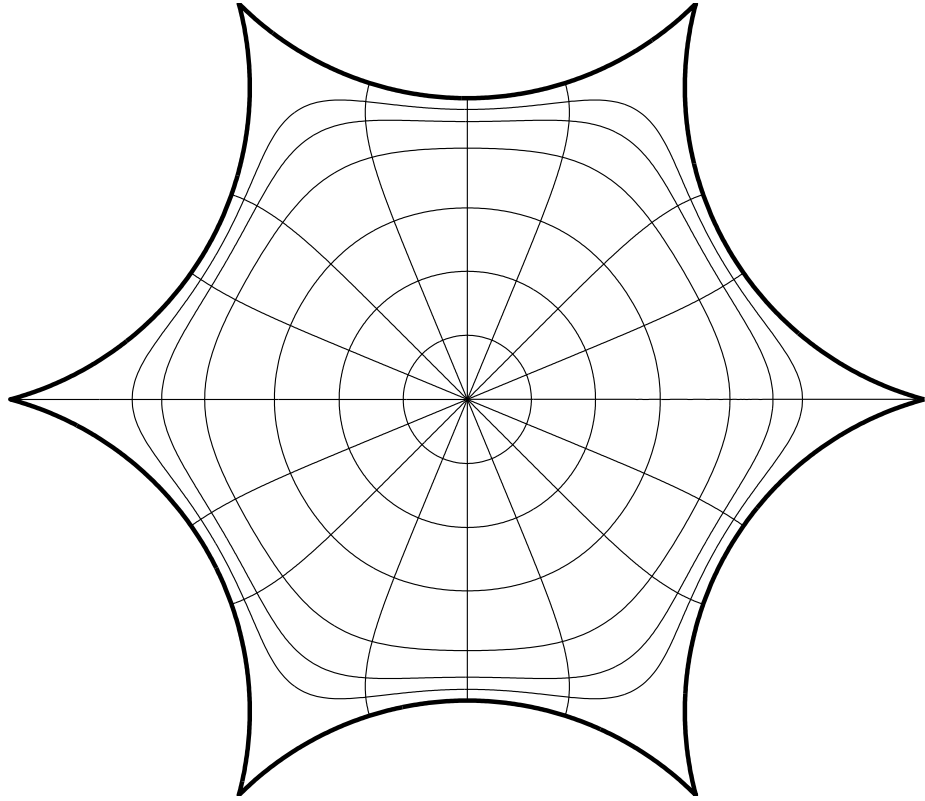


Figure 18: A domain with piecewise smooth boundary has been approximated by a polygon with 250 sides approximating each curved subarc. The curves are the images of 16 equally spaced radii in the unit disk and of concentric circles of radii 0.2, 0.4, 0.6, 0.8, 0.9 and 0.95.

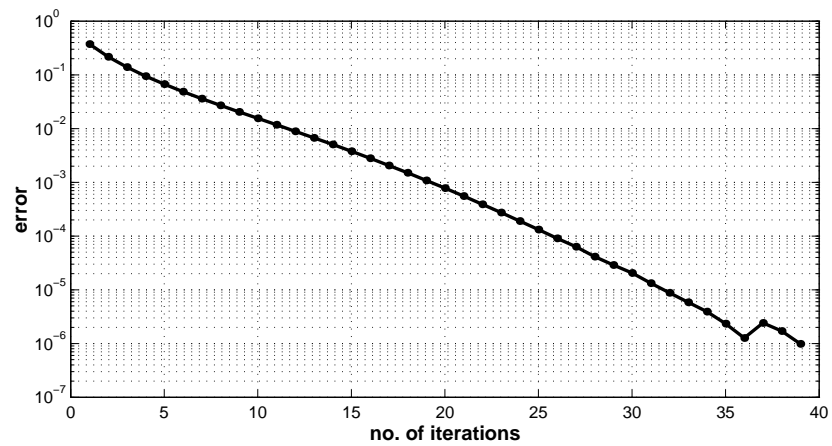


Figure 19: Convergence for the piecewise smooth region in Figure 18.

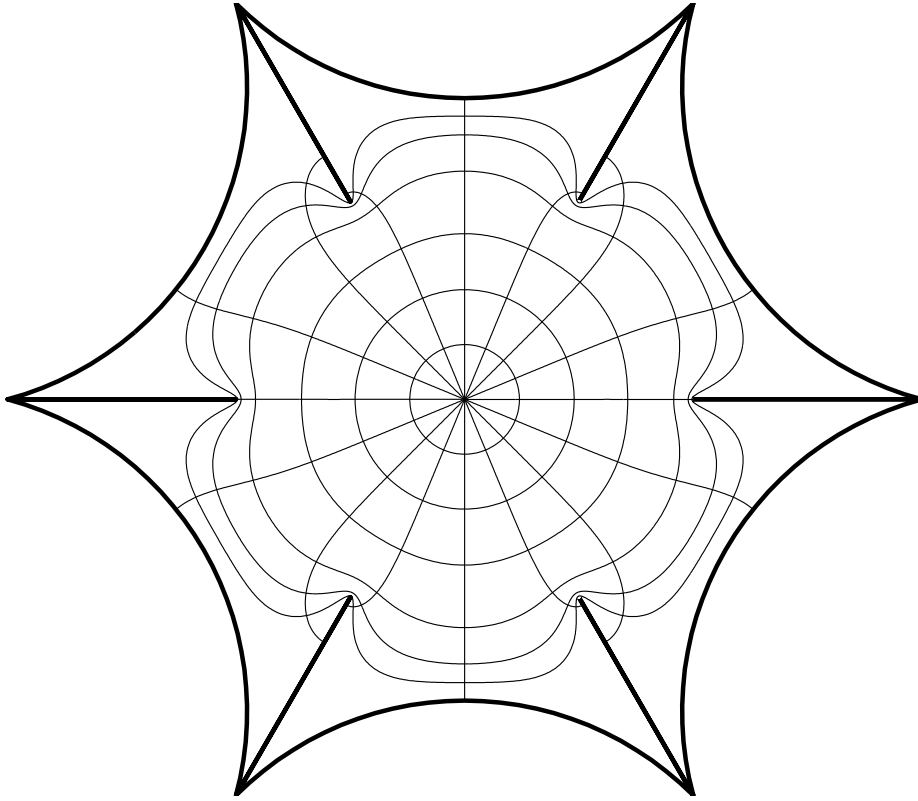


Figure 20: Conformal map from the unit disk to the region from Figure 18 but with added slits. The curves are the images of 16 equally spaced radii in the unit disk and of concentric circles of radii 0.2, 0.4, 0.6, 0.8, 0.9 and 0.95.

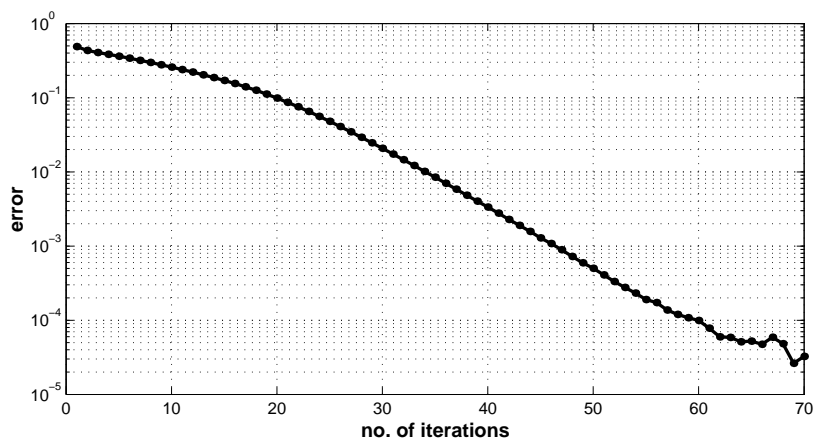


Figure 21: Slower convergence due to the additional slits in the region.

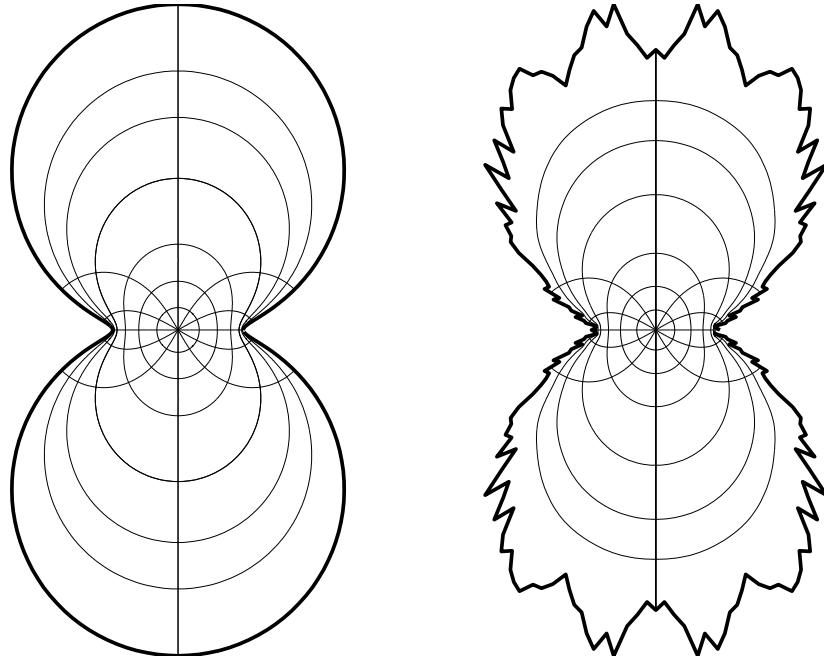


Figure 22: Conformal map from the unit disk to a polygonal approximation to a smooth inverted ellipse and to a distorted inverted ellipse (see Section 19). The curves are the images of 12 equally spaced radii in the unit disk and of concentric circles of radii 0.2, 0.4, 0.6, 0.8, 0.9 and 0.95

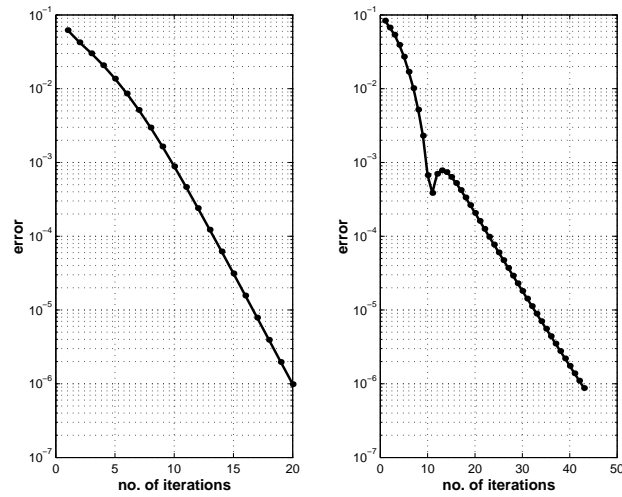


Figure 23: Convergence curves for the same two regions. Note the different horizontal scales.