SIAM J. Sci. Comput.
Vol. 33, No. 3, pp. 1369–1394

NUMERICAL COMPUTATION OF THE SCHWARZ–CHRISTOFFEL TRANSFORMATION FOR MULTIPLY CONNECTED DOMAINS*

THOMAS K. DELILLO† AND EVERETT H. KROPF†

Abstract. We report on recent progress in the computation of Schwarz–Christoffel maps from bounded or unbounded circular domains to conformally equivalent bounded or unbounded multiply connected polygonal domains. The form of the transformation is given in terms of an integral of an infinite product depending on unknown parameters, namely, the prevertices and the centers and radii of the circles. A system of nonlinear equations, which forces the geometry of the given polygonal domain to be correct under the mapping function, is formulated for the unknown parameters and solved by a continuation method. A transformation of the constrained parameters to an unconstrained set of variables is crucial to the effective solution of the system. Several numerical examples are given. The approach here proves to be very robust.

Key words. conformal mapping, Schwarz–Christoffel transformation, multiply connected domains

AMS subject classifications. 30C30, 65E05

DOI. 10.1137/100816912

1. Introduction.

A Schwarz–Christoffel formula for conformal maps $w = f(z)$ from the exterior of a finite number of disks in the z-plane to the exterior of polygonal curves in the w-plane was derived in [17] using the Schwarz reflection principle. A similar formula for the bounded case was derived in [14]. Both of these formulas express the derivative $f'(z)$ of the mapping function in terms of infinite products which depend on the prevertices and the centers and radii of the circles. Once these parameters are found for a given polygonal domain, the map can be computed by numerically integrating $f'(z)$. A very effective method for solving this *parameter problem* for simply connected domains was first developed in [27]. A Fortran package was developed and later redone in MATLAB in [19]. A package for doubly connected Schwarz–Christoffel maps was developed in [23]. However, little has been done for domains of higher connectivity. The book [20] gives an introduction to Schwarz–Christoffel maps and their computation; see also [22] for a broad overview of theory and numerical methods for conformal mapping.

In this paper, we will discuss some recent progress in the numerical implementation of the formulas for higher connectivity derived in [14, 17]. Earlier work on the computation of maps for the unbounded (exterior) case was presented in [15]. This paper significantly improves on the code developed there and extends it to the bounded (interior) case. The key improvement here is the change of the unknown parameters from constrained to unconstrained variables adapting the transformation [3, eq. (21)] to the multiply connected case. (This transformation is a slight revision of the transformation used originally in [27], which is also discussed in [20, p. 25] and adapted to similar problems in [2, eq. (5.5)] and [21, eq. (2.22)].) In [27], a nonlinear system of equations was set up which forces the integral of $f'(z)$ between prevertices to give the correct side lengths for the polygon. In [15], a similar system was set up

*Submitted to the journal's Methods and Algorithms for Scientific Computing section December 3, 2010; accepted for publication (in revised form) April 12, 2011; published electronically June 7, 2011.
http://www.siam.org/journals/sisc/33-3/81691.html

†Department of Mathematics and Statistics, Wichita State University, Wichita, KS 67260-0033 (delillo@math.wichita.edu, kropf@math.wichita.edu).

which forces the side lengths of all the polygons to be correct and also forces positions and orientations of the polygons to be correct. The resulting system is solved by a continuation method. In [15], very accurate initial guesses were needed for all but the simplest of domains in order to get the continuation method to converge to a correct solution. Here the same nonlinear system is solved by the same continuation method, but the change to unconstrained variables results in a very robust code that nearly always converges rapidly to the correct map. The advantage of the unconstrained variables is that they prevent prevertices from crossing and getting out of order. (Such crossings were apparently the main problem with [15], not the problem of integration paths between circles crossing into the circles or through singularities, as we had previously thought.) The approach here can be used to compute bounded interior maps with a slight modification of the nonlinear system which allows the map to be normalized by fixing one interior and one boundary point. We believe the present code provides a useful tool for applications as it stands and sets the stage for many possible future improvements and extensions.

Schwarz–Christoffel formulas for bounded and unbounded multiply connected domains were also developed by Crowdy [7, 8] using the Schottky–Klein prime functions. The relation of Crowdy's formulas to the formulas used here is discussed in [14]. These formulas also depend on the prevertices and the centers and radii of the circles. Some remarks on their potential numerical use will be made in section 7.

This paper is organized as follows. In section 2, we state some preliminaries on reflections and recall the mapping formulas. In section 3, we state the parameter problem and the system of nonlinear equations. In section 4, we derive the transformations to unconstrained variables. In section 5, we briefly review the continuation methods for solving the nonlinear systems. In section 6, we give a number of numerical examples demonstrating the robustness of the method. In section 7, we summarize our results and suggest directions for future research.

2. Preliminaries.

Here we recall the notation and formulas from [14, 17] where infinite product formulas for $f'(z)$ for the unbounded ((2.2) below) and bounded ((2.3) below) cases are given in terms of repeated reflections of prevertices and circle centers. Sufficient conditions for convergence of these infinite products are given in [14, 17]. These conditions are far from necessary in practice. We will only recall enough details here so that the reader can understand the formulas we are implementing, the violation of the sufficiency conditions in many of our examples, and a better method in practice for estimating the convergence rate and error in the truncated infinite products used in our computations. We emphasize here that the existence of the conformal maps $w = f(z)$ from the circle domains to the polygonal domains is guaranteed by theorems on conformal mapping for general domains, such as [22, Theorem 17.6a]. The purpose of the Schwarz–Christoffel formula is to give a specific (and computationally useful) form of f for polygonal domains.

2.1. Reflections in circles and notation.

We need some basic facts about reflections in circles and some useful lemmas from [14, 17]. We first state some notation. The conformal map from the (bounded or unbounded) circle domain Ω to the (bounded or unbounded) polygonal domain $\mathbb{P}$ of *connectivity* m is denoted by $w = f(z)$. For $i = 1, \dots, m$, c_i and r_i are the centers and radii of the (nonintersecting) circles, C_i. C_1 is the unit circle with $c_1 = 0, r_1 = 1$ for both the bounded and unbounded case. For the unbounded case, the circles are mutually exterior. For the bounded case, the circles $C_2, \dots, C_m$ are in the interior of C_1. (Note that in [14] the outer unit circle is denoted C_0 and the connectivity is $m+1$.) The *vertices* (or corners) of the polygon are $w_{k,i} \in \Gamma_i$, the ith polygon, with $k = 1, \ldots, K_i$, and $w_{k,i} = f(z_{k,i})$ for the *prevertices*, $z_{k,i} = c_i + r_i e^{i\theta_{k,i}} \in C_i$. In addition, $\alpha_{k,i}\pi$ are the interior angles of the polygons at the corners, $w_{k,i}$, and $\beta_{k,i}\pi, i = 1, \ldots, m$, are the *turning angles* of the tangent at $w_{k,i}$ with $\beta_{k,i} = \alpha_{k,i} - 1$. (The $-\beta_{k,1}\pi$'s are the turning angles for the outer polygon in the bounded case.)

Next, we give some details of our notation for reflections. We first recall that *reflection* of a point z through a circle C_τ with center c_τ and radius r_τ is given by

$$z_\tau = \rho_\tau(z) := c_\tau + \frac{r_\tau^2}{\overline{z} - \overline{c_\tau}},$$

i.e., z and z_τ are *symmetric points with respect to the circle* C_τ. (Note that if $z \in C_\tau$, then $z_\tau = z$.) We will define the following sets of multi-indices to label sequences of repeated reflections.

Definition 2.1. *The set of multi-indices ν of length $|\nu| = n$ is denoted*

$$\sigma_n = \{\nu_1\nu_2 \cdots \nu_n : 1 \leq \nu_j \leq m, \nu_k \neq \nu_{k+1}, k = 1, \ldots, n-1\}, \quad n > 0,$$

and $\sigma_0 = \phi$. (If $\nu \in \sigma_0$, then $\nu i = i$.) Also

$$\sigma_n(i) = \{\nu \in \sigma_n : \nu_n \neq i\},$$

denotes sequences in σ_n whose last factor never equals i.

For example, for $m = 3$, $\sigma_3 = \{121, 123, 131, 132, 212, 213, 231, 232, 312, 313, 321, 323\}$ and $\sigma_3(1) = \{123, 132, 212, 213, 232, 312, 313, 323\}$.

Our Schwarz–Christoffel formulas (2.2) and (2.3) are based on analytic extension of the mapping function $w = f(z)$ by Schwarz reflection across the arcs of circles between prevertices $z_{k,i}$ and the corresponding sides of the polygons. The reflection process is repeated across the reflected circles and polygons an infinite number of times until the entire plane of the circle domain (minus some limit set) is covered. The complete mathematical and notational details of the reflection process is given in [17]. We will give a summary and some examples here, especially in order to explain [17, sect. 2.1] and our reflection algorithm. Consider the unbounded case with $m = 3$. The reflections of the circles C_2 and C_3 through circle C_1 are denoted $C_{12} = \rho_1(C_2)$ and $C_{13} = \rho_1(C_3)$, respectively, and function $w = f(z)$ is extended to the domain bounded by $C_1 \cup C_{12} \cup C_{13}$ by Schwarz reflection of z through an arc on C_1 between two successive prevertices and w through the corresponding side of the polygon. If another arc between two successive vertices on C_1 is chosen, the same reflected z will correspond to a different value of $w = f(z)$ obtained by reflection through a different side of the polygon. The extension of $f(z)$ will therefore be multivalued. However, as is shown in [17], the pre-Schwarzian, $S(z) = f''(z)/f'(z)$, is single-valued. The reflection process is continued with, for instance, the circle C_1 reflected through C_{12} to $C_{121} = \rho_{12}(C_1)$ and C_{13} reflected through C_{12} to $C_{123} = \rho_{12}(C_{13}) = \rho_{12}(\rho_1(C_3))$, etc. An arbitrary reflected circle is denoted by C_ν with a multi-index ν, defined above, labeling the sequence of reflections. Some care is required to reconstruct the original sequence of reflections from the index ν, as illustrated by C_{123}. However, note that, e.g., $C_\nu = C_{\nu_1\nu_2\cdots\nu_n}$ is in the interior of C_{ν_1} and arises from a sequence of reflections of C_{ν_n}; see also, the remark at the end of this subsection.

The reflections of prevertices themselves are singularities (or branch points or zeros) of the extended function and are needed in the formulas. These reflections are denoted by $z_{k,\nu i}$. For instance, for $z_{k,2} \in C_2$, the kth prevertex on C_2, its reflection through C_1 is denoted $z_{k,12} = \rho_1(z_{k,2}) \in C_{12}$. In general, $z_{k,\nu} \in C_\nu$. For the unbounded case, the simple pole at infinity is also reflected and shows up as repeated reflections, $s_{\nu i}$, of the centers, $s_i := c_i$. For instance, the reflection of $s_2 := c_2$, the center of circle C_2, through C_1 is denoted s_{12}. (Note that, e.g., $s_{12} = \rho_1(c_2) = \rho_1(\rho_2(\infty)) \neq c_{12} = \rho_{12}(\infty)$, the center of circle C_{12}.) By the basic theorem for the existence of the mapping function, given a polygonal domain, the circles are uniquely determined up to a normalization and, therefore, the set of all reflections is also determined.

It is a very useful fact for our computations that the set of all reflections through our circles $C_i, i = 1, \ldots, m$, and through reflections of the circles, etc., can be generated entirely by compositions of reflections in the original circles, C_i. Note that the reflections do not commute, so that the order in which they are carried out is important. That is, given two circles C_i and C_j, if $i \neq j$, then $\rho_i(\rho_j(a)) \neq \rho_j(\rho_i(a))$. However, if the reflection of circle j through circle i is denoted $C_{ij} = \rho_i(C_j)$, then [17, Prop. 1] says that

$$\rho_i(\rho_j(a)) = \rho_{ij}(\rho_i(a)). \tag{2.1}$$

The proof of (2.1) from [17] is easy to see. Recall that Moebius transformations preserve reflections in circles and straight lines, and let C_i be the real axis, where reflection is just complex conjugation. Therefore,

$$\rho_i(\rho_j(a)) = \overline{\rho_j(a)} = \rho_{ij}(\overline{a}) = \rho_{ij}(\rho_i(a)).$$

Repeated use of (2.1) shows that reflection through any reflected circle, C_ν, can be factored into repeated reflections through the C_i's. This is expressed in the following "improved" form of [17, Lemma 1] and is the key to our notation and our computations of reflections.

Lemma 2.2. *For $\nu = \nu_1\nu_2\cdots\nu_n \in \sigma_n(i)$,*

$$\begin{aligned} C_{\nu i} &= \rho_{\nu_1}(\rho_{\nu_2}(\cdots(\rho_{\nu_n}(C_i))\cdots)), \\ z_{k,\nu i} &= \rho_{\nu_1}(\rho_{\nu_2}(\cdots(\rho_{\nu_n}(z_{k,i}))\cdots)), \quad \text{and} \\ s_{\nu i} &= \rho_{\nu_1}(\rho_{\nu_2}(\cdots(\rho_{\nu_n}(s_i))\cdots)). \end{aligned}$$

For example, $C_{21} = \rho_2(C_1)$ and $z_{k,21} = \rho_2(z_{k,1})$. The proof of [17, Lemma 1] uses (2.1) and an induction argument, which we will not repeat, to show that, e.g., $C_{\nu i} = \rho_{\nu_1}(C_{\nu_2\cdots\nu_n i})$. Repeated application of this to ν_2 through ν_n gives our Lemma 2.2.

Based on Lemma 2.2, we have developed a MATLAB code which performs the reflections of centers and prevertices to a specified level $|\nu| = N$. Only reflections across the original circles are computed. For instance, $s_{123} = \rho_{12}(s_{23})$ is computed as $s_{123} = \rho_1(\rho_2(s_3))$. The code stores an expanding array of integers $j \in \{1, \ldots, m\}$, for each reflection such that ρ_j is the most recent reflection. For the next level of reflections, ρ_j is then skipped, since $\rho_j(\rho_j(a)) = a$. A listing of a simplified version of the code is given in [18, Alg. 2.2, p. 198] and can be viewed online. The code also computes the centers and radii of the reflected circle which are needed for some of our calculations below. (For the bounded case, the $m - 1$ inner circles are initially reflected to the exterior of the (dashed) unit circle.)

Figure 2.1 illustrates the reflections of circles in circles for the bounded and unbounded cases and their truncation after one level. The dashed circle is always the

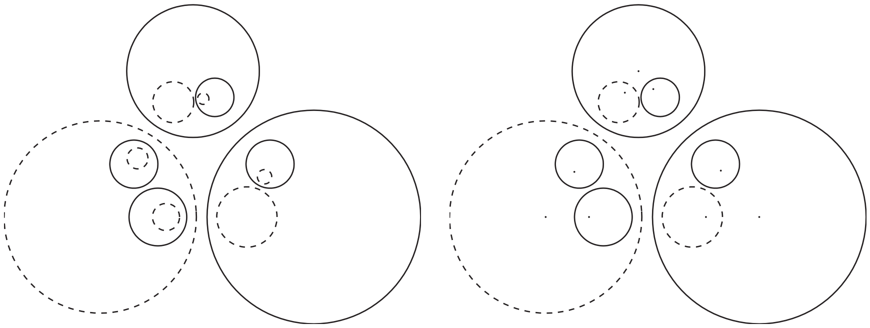

FIG. 2.1. *$N = 1$ levels of reflected circles for the bounded case (left) and the unbounded case (right). The dashed circles are the unit circle and its reflections. The dots are the centers of the outer circles and their reflections. For the bounded case, each reflection of a solid circle is paired with a reflection of the (dashed) unit circle. For the unbounded case, each reflection of any circle is paired with a corresponding reflection of its center.*

unit circle and is the outer computational circle in the bounded case. For our computations, the infinite product formulas for $f'(z)$ must be truncated to a level of $|\nu| = N$ reflections. The truncated products are given below in (3.2) for the unbounded case and (3.3) for the bounded case. Note that for the unbounded case a reflection of a center is always paired with a reflected circle. Correspondingly, in the bounded case a reflection of the unit circle is always paired with a reflected (solid) circle, $C_i, i \neq 1$. (This is necessary in truncating the products because it ensures that the total 2 (or -2) residues of the truncated singularity function, $S_N(z) \approx f''(z)/f'(z)$ discussed below, accompanying the circles—center reflections or reflections of outer circles in the bounded case—always sum to zero, thus yielding zero periods and hence single-valuedness when integrating around circles.)

Remark: We wish to clarify some ambiguities of notation in [14, 15, 18]. In those papers, we denote, e.g., $\rho_\nu(a) := \rho_{\nu_1}(\rho_{\nu_2}(\cdots(\rho_{\nu_n}(a))\cdots))$, so $\rho_\nu(a)$ should not be read as the reflection of a through C_ν, especially in our infinite product formulas for slit maps. Probably a_ν would have been more consistent notation than $\rho_\nu(a)$.

2.2. Schwarz–Christoffel formulas for multiply connected domains. We discuss formulas for Schwarz–Christoffel maps from bounded or unbounded circular domains to bounded or unbounded polygonal domains. Recall the framework of [20] wherein the derivative of the mapping function, $f'(z)$, is expressed as a product $f' = A\prod_k f_k$ of factors $f_k(z)$ that yield the correct behavior of the map at the kth corner for the given geometry. For instance, for the case of simply connected maps from the disk, the derivative is

$$f'(z) = A\prod_k f_k(z) = A\prod_k (z - z_k)^{\beta_k},$$

where $f_k(z) := (z - z_k)^{\beta_k}$, $-\beta_k\pi$ is the turning angle at prevertex z_k, $\beta_k = \alpha_k - 1$, and $\sum_k \beta_k = -2$. In this case, the mapping function is

$$f(z) = A\int^z \prod_k (\zeta - z_k)^{\beta_k} d\zeta + B,$$

where a normalization condition, such as fixing an interior point and one boundary point, gives a unique map. (The numerical problem in this case is to find A, B, z_k's by matching side lengths of the polygon.)

To set the context for the paper, we will first review the existing formulas for multiply connected Schwarz–Christoffel maps (MCSC). The MCSC formula for the unbounded case [17] is

$$f'(z) = A \prod_{i=1}^{m} \prod_{k=1}^{K_i} \prod_{\substack{j=1 \\ \nu \in \sigma_j(i)}}^{\infty} \left(\frac{z - z_{k,\nu i}}{z - s_{\nu i}} \right)^{\beta_{k,i}}, \tag{2.2}$$

where $z_{k,\nu i}$'s are reflections of the prevertices $z_{k,i}$ of the kth prevertex on the ith circle, $s_{\nu i}$ are reflections of the centers $s_i = c_i$ of the ith circle, and the turning angles $\beta_{k,i}\pi$ at each vertex satisfy $\sum_{k=1}^{K_i} \beta_{k,i} = 2, i = 1, \ldots, m$. In this case $f(z)$ has a simple pole at $z \approx \infty$. Since ∞ is the reflected to the circle centers c_i, the extension of f by Schwarz reflection will have singularities at the reflections of the centers and the prevertices. The corresponding MCSC formula for the bounded case from [14] is

$$f'(z) = A \prod_{k=1}^{K_1} (z - z_{k,1})^{\beta_{k,1}} \prod_{i=2}^{m} \prod_{\substack{j=0 \\ \nu \in \sigma_j(i)}}^{\infty} \left(\prod_{k=1}^{K_1} (z - z_{k,\nu i1})^{\beta_{k,1}} \prod_{k=1}^{K_i} (z - z_{k,\nu i})^{\beta_{k,i}} \right), \tag{2.3}$$

where $z_{k,1}$ are prevertices on the outer circle, taken to be the unit circle, and $\sum_{k=1}^{K_1} \beta_{k,1} = -2, \sum_{k=1}^{K_i} \beta_{k,i} = 2, i = 2, \ldots, m$. In this case, the map $f(z)$ has no pole in its domain and the reflections of the prevertices on the outer circle, in effect, replace the reflections of the centers in the formula for the unbounded case.

The derivations use the following lemma, which says that the arcs between prevertices map to straight sides.

Lemma 2.3. $\operatorname{Re}\{(z - c_j) f''(z)/f'(z)\}_{|z-c_j|=r_j} = -1$, $j = 1, \ldots, m$.

We also define

$$\Delta := \max_{i,j;i \neq j} \frac{r_i + r_j}{|c_i - c_j|} < 1, \quad 1 \leq i,j \leq m,$$

the *separation parameter* of the circle domain. If $\widetilde{C}_j$ denotes the circle with center c_j and radius r_j/Δ, then geometrically $1/\Delta$ is the smallest magnification of the m radii such that at least two $\widetilde{C}_j$'s just touch. The theorem for the unbounded formula is as follows.

Theorem 2.4. *If the unbounded m-connected circular domain Ω satisfies the separation property $\Delta < (m-1)^{-1/4}$ for $m > 1$, then the Schwarz–Christoffel map to the polygonal domain $\mathbb{P}$ is*

$$f(z) = A \int^{z} \prod_{i=i}^{m} \prod_{k=1}^{K_i} \left[\prod_{\substack{j=1 \\ \nu \in \sigma_j(i)}}^{\infty} \left(\frac{\zeta - z_{k,\nu i}}{z - s_{\nu i}} \right) \right]^{\beta_{k,i}} d\zeta + B,$$

where $-1 < \beta_{k,i} \leq 1$ and $\sum_{k=1}^{m} \beta_{k,i} = 2$.

Note that $\Delta < (m-1)^{1/4}$ is a sufficient condition for convergence. It is not necessary and is violated in many of our numerical examples.

Proof. We give only enough details of the convergence proofs in [17] for the reader to understand our error estimate. The bounded case [14] is similar. The central idea is to prove that the pre-Schwarzian $f''(z)/f'(z) = S(z)$ by means of the argument principle, where $S(z)$ is the singularity function, constructed from the poles of the pre-Schwarzian. The pre-Schwarzian has simple poles at the $z_{k,\nu i}$ with residue $\beta_{k,i}$ and at $s_{\nu i}$ with residue -2. We define $S(z)$ by writing, for $j = 0, 1, 2, \ldots,$

$$A_j\,(z) = \sum_{i=1}^{m} \sum_{\nu\in\sigma_j(i)} \left(\sum_{k=1}^{K_i} \frac{\beta_{k,i}}{z - z_{k,\nu i}} - \frac{2}{z - s_{\nu i}} \right) = \sum_{i=1}^{m} \sum_{\nu\in\sigma_j(i)} \sum_{k=1}^{K_i} \frac{\beta_{k,i}(z_{k,\nu i} - s_{\nu i})}{(z - z_{k,\nu i})(z - s_{\nu i})}$$

and $S_N(z) := \sum_{j=0}^{N} A_j(z)$. Therefore, when the series converges, we define

$$S(z) := \lim_{N\to\infty} S_N(z).$$

(In addition to the convergence of $S_N(z)$, it is also necessary to show that

$$\operatorname{Re}\{(z - s_j)S(z)\}_{z\in C_j} = -1, \quad j = 1, \ldots, m,$$

so that $S(z)$ satisfies the boundary conditions of Lemma 2.3 for the pre-Schwarzian. However, we have no need to discuss the details here.)

Finally, since $S(z) = d\log f'(z)/dz$, we have

$$A\exp\left(\int S_N(z)dz\right) \to f'(z) = A\exp\left(\int S(z)dz\right) = A\prod_{i=0}^{m}\prod_{k=1}^{K_i}\prod_{\substack{j=0\\ \nu\in\sigma_j(i)}}^{\infty} \left(\frac{z - z_{k,\nu i}}{z - s_{\nu i}}\right)^{\beta_{k,i}}.$$

Note that the number of terms in the $A_j(z)$ sum is $O((m-1)^j)$. This exponential increase in the number of terms is the principal difficulty in establishing convergence. Let $r_{\nu i}$ be the radius of $C_{\nu i}$, the νth reflection of circle C_i. We bound $A_j(z)$, for $z \in \Omega$ bounded away from the $z_{k,\nu}$'s by δ, using $-1 < \beta_{k,i} \le 1$, $|z_{k,\nu i} - s_{\nu i}| < 2r_{\nu i}$, $K_{\max} := \max\limits_i K_i$, and the Cauchy–Schwarz inequality, as follows:

$$(2.4) \qquad |A_j(z)| \le \sum_{\nu\in\sigma_j(i)} \sum_{i=1}^{m} \sum_{k=1}^{K_i} \frac{|\beta_{k,i}||z_{k,\nu i} - s_{\nu i}|}{|z - z_{k,\nu i}||z - s_{\nu i}|} \le \frac{2K_{\max}}{\delta^2} \sum_{\nu\in\sigma_j(i)} \sum_{i=1}^{m} r_{\nu i}.$$

We will see below that the sum of the radii $r_{\nu i}$ at the $j = N$th level of reflection gives a much better estimate of the truncation error than the following estimate leading to our sufficient condition. To estimate the rate of decrease of the radii and prove convergence, we need to estimate how fast the reflected circles shrink. This is done in [14, 17] with the following lemma.

Lemma 2.5 (see [22, p. 505]).

$$\sum_{\nu\in\sigma_{n+1}} r_\nu^2 \le \Delta^{4n} \sum_{i=1}^{m} r_i^2.$$

Applying Lemma 2.5 to $\sum r_{\nu i}$ above, we have

$$\sum_{\nu\in\sigma_j(i)} \sum_{i=1}^{m} r_{\nu i} \le \left(\sum_{\nu\in\sigma_j(i)} \sum_{i=1}^{m} r_{\nu i}^2 \right)^{1/2} \left(\sum_{\nu\in\sigma_j(i)} \sum_{i=1}^{m} 1 \right)^{1/2}$$

$$\le \Delta^{2j} \left(\sum_{i=1}^{m} r_i^2 \right)^{1/2} \sqrt{m}(m-1)^{j/2} \le C\Delta^{2j}(m-1)^{j/2}.$$

Therefore, the series converges if $\Delta^2\sqrt{m-1} < 1$. □

Our MATLAB code produces the reflections of centers and prevertices to a fixed level, along with the centers of the reflected circles, $c_{\nu i}$. Our product formula uses input from this code. In order to keep on the principal branches of the $\beta_{k,i}$ roots when the variable of integration ζ is in the computational domain, the bounded Schwarz–Christoffel formula is expressed as

$$\begin{aligned} f\left(z\right) &= A\int^{z} \prod_{i=1}^{m}\prod_{k=1}^{K_i} \prod_{\substack{j=0\\ \nu\in\sigma_j(i)}}^{\infty} \left(\zeta - z_{k,\nu i}\right)^{\beta_{k,i}}\, d\zeta + B \\ &= A'\int^{z} \prod_{k=1}^{K_1}\left(1-\frac{\zeta}{z_{k,1}}\right)^{\beta_{k,1}} \prod_{i=2}^{m}\prod_{k=1}^{K_i} \prod_{\substack{j=0\\ \nu\in\sigma_j(i)}}^{\infty} \left(1-\frac{z_{k,\nu i}-c_{\nu i}}{\zeta - c_{\nu i}}\right)^{\beta_{k,i}} \\ &\quad\times \prod_{k=1}^{K_1} \prod_{\substack{j=0\\ \nu\in\sigma_j(i)}}^{\infty} \left(1-\frac{z_{k,\nu i1}-c_{\nu i}}{\zeta - c_{\nu i}}\right)^{\beta_{k,1}} d\zeta + B. \end{aligned}$$

(We use a similar expression for the unbounded case.)

3. Parameter problem.

The Schwarz–Christoffel transformations above give the form of the mapping functions f for polygonal domains $\mathbb{P}$. Therefore, in order to apply these formulas to the computation of the mapping function, we need to find the prevertices $z_{k,i}$, such that $f(z_{k,i}) = w_{k,i}$, along with the centers c_i and radii r_i of the conformally equivalent circle domain. Recall that K_i for $i = 1, \ldots, m$ are the number of vertices on each Γ_i in $\partial\mathbb{P}$. Parametrize the prevertices by $z_{k,i} = c_i + r_i e^{i\theta_{k,i}}$ for $k = 1, \ldots, K_i$ with

$$\theta_{1,i} < \theta_{2,i} < \cdots < \theta_{K_i,i}. \tag{3.1}$$

The unknown c_i's, r_i's, and $\theta_{k,i}$'s amount to a total of

$$K_1 + \cdots + K_m + 3m$$

real parameters. This is exactly the number of real parameters needed to determine the Schwarz–Christoffel product for $f'(z)$ in both the bounded and unbounded cases. We truncate the infinite products to N levels for the unbounded case,

$$p_u(z) = \prod_{i=1}^{m}\prod_{k=1}^{K_i} \prod_{\substack{j=0\\ \nu\in\sigma_\mu(i)}}^{N} \left(\frac{z - z_{k,\nu i}}{z - s_{\nu i}}\right)^{\beta_{k,i}} \approx \prod_{i=1}^{m}\prod_{k=1}^{K_i} \prod_{\substack{j=0\\ \nu\in\sigma_\mu(i)}}^{\infty} \left(\frac{z - z_{k,\nu i}}{z - s_{\nu i}}\right)^{\beta_{k,i}}, \tag{3.2}$$

and the bounded case,

$$p_b(z) = \prod_{k=1}^{K_1}\left(z - z_{k,1}\right)^{\beta_{k,1}} \prod_{i=2}^{m} \prod_{\substack{j=0\\ \nu\in\sigma_j(i)}}^{N} \left(\prod_{k=1}^{K_1}\left(z - z_{k,\nu i1}\right)^{\beta_{k,1}} \prod_{k=1}^{K_i}\left(z - z_{k,\nu i}\right)^{\beta_{k,i}} \right). \tag{3.3}$$

We will apply the normalization conditions for conformal maps of multiply connected domains and enforce conditions based on the geometry of the polygonal domains to set up (complete) systems of (independent) nonlinear equations for these unknown parameters. The choice of the geometric conditions is not unique, as we will demonstrate, and the unbounded case is slightly different than the bounded case, due to the different normalizations.

3.1. Unbounded case. The nonlinear equations for this case were first given in [15]. We recall them here. The theory of conformal maps for multiply connected domains [22, 26] shows that, given $\mathbb{P}$ and the normalization condition, $f(z) = z + O(1/z)$ for $z \to \infty$, there is a unique circle domain Ω such that f uniquely maps Ω onto $\mathbb{P}$. Since this normalization is difficult to impose, it is relaxed to

$$f(z) = Az + B + O(1/z), \quad z \to \infty,$$

and A and B are determined implicitly. To do this, we set $c_1 = 0$, $r_1 = 1$, and $\theta_{1,1} = 0$. We define

$$C = \frac{w_{2,1} - w_{1,1}}{\int_{z_{1,1}}^{z_{2,1}} p_u(\zeta)\, d\zeta}$$

and then write

$$f(z) = C \int_{z_{1,1}}^{z} p_u(\zeta)\, d\zeta + D,$$

with $D = w_{1,1} = f(z_{1,1})$. This normalization takes care of 4 of the real parameters, leaving

$$(K_1 - 1) + K_2 + \cdots + K_m + (3m - 3) = K_1 + \cdots + K_m + 3m - 4$$

real parameters to be determined.

The nonlinear conditions are derived from the geometry of the polygonal domain. To ensure the correct side lengths of the polygons, we have the *side-length conditions*,

$$|f(z_{k+1,i}) - f(z_{k,i})| = |w_{k+1,i} - w_{k,i}|,$$

for $i = 1, \ldots, m$ and $k = 1, \ldots, K_i$, with $z_{K_i+1,i} := z_{1,i}, w_{K_i+1,i} := w_{1,i}$. This is a total of $K_1 + \cdots + K_m$ equations, but the definition of C above fixes the first side length of the first polygon, which removes one from this count. The *positions* of polygons Γ_2 through Γ_m in relation to Γ_1 (the position of the first polygon Γ_1 is fixed by the normalization) are determined by

$$f(z_{1,i}) - f(z_{1,1}) = w_{1,i} - w_{1,1}$$

for $i = 2, \ldots, m$, which gives $2(m-1)$ real equations. Finally, the *orientations* of polygons Γ_2 through Γ_m are determined by

$$\arg(f(z_{2,i}) - f(z_{1,i})) = \arg(w_{2,i} - w_{1,i})$$

for $i = 2, \ldots, m$ (again the orientation of Γ_1 is determined by the normalization), which gives $(m-1)$ real equations. These equations can be combined with the side-length conditions for $k = 1$ to obtain

$$f(z_{2,i}) - f(z_{1,i}) = w_{2,i} - w_{1,i}, \qquad i = 2, \ldots, m.$$

This set of conditions amounts to a system of $K_1 + \cdots + K_m + 3m - 4$ equations for an equal number of unknowns.

3.2. Bounded case. In this case, in order to guarantee uniqueness of the mapping function, the map can be normalized by fixing one boundary point, $f(1) = w_{1,1}$, and one interior point, $f(z_0) = w_0$, where z_0 is in the interior of the circle domain and w_0 is in the interior of the polygonal domain. (The map can also be normalized by fixing three vertices on the outer boundary, but we will not implement this here.) Again letting

$$C = \frac{w_{2,1} - w_{1,1}}{\int_{z_{1,1}}^{z_{2,1}} p_b(z)\, dz},$$

we have

$$f(z) = C \int_{z_{1,1}}^{z} p_b(\zeta)\, d\zeta + D,$$

with $f(z_{1,1}) = D = w_{1,1}$. We again require that $c_1 = 0$ and $r_1 = 1$, and fixing $f(1) = w_{1,1}$ is equivalent to setting $\theta_{1,1} = 0$. This amounts to fixing four of the real parameters, so once again we have $K_1 + \cdots + K_m + 3m - 4$ unknowns parameters to determine.

The remaining parameters are determined as follows. As in the bounded case we have the *side-length conditions*,

$$|f(z_{k+1,i}) - f(z_{k,i})| = |w_{k+1,i} - w_{k,i}|$$

for $i = 1, \ldots, m$ and $k = 1, \ldots, K_i$. These are $K_1 + \cdots + K_m$ real equations, but the calculation of C again removes one from this count. In addition, we leave off the calculation of the last two side lengths of the outer boundary polygon ($i = 1$); this works since the known turning angles of the polygon allow the last vertex to be uniquely determined by the intersection of lines drawn from the adjacent vertices; see [20, Fig. 3.1, p. 24]. The side-length conditions then add up to $K_1 + \cdots + K_m - 3$ real equations. The *positions* of Γ_1 through Γ_m with respect to w_0 are given by

$$f(z_{1,i}) - f(z_0) = w_{1,i} - w_0$$

for $i = 1, \ldots, m$. These position conditions give $2m$ real equations. Finally, the *orientations* of Γ_2 through Γ_m (the orientation of Γ_1 is determined by the calculation of C) are given by the $(m - 1)$ real equations,

$$\arg(f(z_{2,i}) - f(z_{1,i})) = \arg(w_{2,i} - w_{1,i})$$

for $i = 2, \ldots, m$. Once again, the side-length, position, and orientation conditions give

$$K_1 + \cdots + K_m + 3m - 4$$

real equations, exactly as needed.

4. Transformation to unconstrained variables.

The constraint (3.1) is difficult to enforce directly on the $\theta_{k,j}$'s, and its violation can cause the nonlinear solver to fail to converge to a good solution. Therefore, as in [3, 20, 27], we make a transformation to a set of unconstrained variables which guarantees (3.1). This change was not implemented in the initial attempt to compute multiply connected Schwarz–Christoffel maps in [15], and it severely limited the robustness of the algorithm. With this change, our method is extremely robust and rarely fails, with very little care needed in the selection on an initial guess. Fortunately, it does not seem to be necessary, in practice, to constrain the circles from overlapping by enforcing $\Delta < 1$. This would be a difficult set of conditions to apply.

We use the unconstrained variables, $\mathrm{Re}\{c_i\}, \mathrm{Im}\{c_i\}$, and $\log r_i$ in place of the centers and radii. For the $\theta_{k,i}$'s (denoted here by θ_k), the following transformation [3, eq. (21)] is used. On any of the boundary circles in the computational domain with K prevertices, we have the problem of finding the angles of these prevertices with respect to the circle center. The angles, of course, must meet the constraint,

$$\sum_{k=1}^{K}(\theta_{k+1}-\theta_k) = 2\pi,$$

where we use the convention $\theta_{K+1} = \theta_1 + 2\pi$. Denoting

$$\phi_k := \theta_{k+1} - \theta_k \quad \text{for } k = 1, \ldots, K,$$

the unconstrained variables (with $\psi_{k,j} = \psi_k$) for a given circle are

$$\psi_k := \log \frac{\phi_{k+1}}{\phi_1} \quad \text{for } k = 1, \ldots, K-1. \tag{4.1}$$

In order to invert this transformation, note that

$$1 + \sum_{j=1}^{K-1} e^{\psi_j} = 1 + \frac{\phi_2}{\phi_1} + \cdots + \frac{\phi_K}{\phi_1} = \frac{2\pi}{\phi_1}$$

and

$$1 + \sum_{j=1}^{k-2} e^{\psi_j} = 1 + \frac{\phi_2}{\phi_1} + \cdots + \frac{\phi_{k-1}}{\phi_1} = \frac{\theta_2-\theta_1}{\phi_1} + \frac{\theta_3-\theta_2}{\phi_1} + \cdots + \frac{\theta_k-\theta_{k-1}}{\phi_1} = \frac{\theta_k-\theta_1}{\phi_1},$$

so that

$$2\pi \frac{1 + \sum_{j=1}^{k-2} e^{\psi_j}}{1 + \sum_{j=1}^{K-1} e^{\psi_j}} = 2\pi \frac{(\theta_k-\theta_1)/\phi_1}{2\pi/\phi_1} = \theta_k - \theta_1.$$

Assume that θ_1 is known for the circle in question. Then any angle θ_k, for $k = 2, \ldots, K$, is given by

$$\theta_k = \theta_1 + 2\pi \frac{1 + \sum_{j=1}^{k-2} e^{\psi_j}}{1 + \sum_{j=1}^{K-1} e^{\psi_j}}. \tag{4.2}$$

Therefore, the transformation (4.1) gives $K_1 - 1$ unconstrained parameters,

$$\psi_{1,1}, \psi_{2,1}, \ldots, \psi_{K_1-1,1},$$

on C_1 (recall $\theta_{1,1}$ is fixed at zero). On each of the other circles with K_i prevertices, $i = 2, \ldots, m$, we allow the angle of the first prevertex $\theta_{1,i}$ to be unconstrained (this angle is easily renormalized to 2π) for exactly K_i unconstrained parameters:

$$\theta_{1,i}, \psi_{1,i}, \psi_{2,i}, \ldots, \psi_{K_i-1,i}.$$

The remaining angles on all circles, $\theta_{2,i},\dots,\theta_{K_i,i}$ for $i=1,\dots,m$, are then recovered by (4.2), and (3.1) is automatically satisfied.

Remark. The transformation [20, p. 25] was also tried and leads to similar results for most of the examples here. A more careful implementation of the ideas in [3], including adaptive quadrature to subdivide integration intervals when other singularities are close [3, 27], will most likely be necessary in cases of crowding.

5. Numerical continuation. The equations above can be expressed as a nonlinear system,

$$F(x)=0, \tag{5.1}$$

where $F:\mathbb{R}^n\to\mathbb{R}^n$, with $n=K_1+\cdots+K_m+3m-4$. As in [15], we use the continuation algorithm Contup from [1, Program 3]. We give a brief description of this algorithm. It is assumed that F is smooth enough; that is, F has enough derivatives to facilitate the required analysis. Let G be a trivial map $G:\mathbb{R}^n\to\mathbb{R}^n$ with known zeros. Let $x_0,x_1\in\mathbb{R}^n$ be such that $G(x_0)=0$ and $F(x_1)=0$. Define the homotopy function

$$H(x,\lambda)=\lambda F(x)+(1-\lambda)G(x)$$

for $\lambda\in[0,1]$. Note then that

$$H^{-1}(0)=\ker H=\{(x,\lambda):H(x,\lambda)=0\}\subset\mathbb{R}^{n+1}.$$

Define a curve $c(s)\in\ker H$, i.e.,

$$c:\mathbb{R}\to\ker H,$$

such that $c(s_0)=(x_0,0)$ and $c(s_1)=(x_1,1)$ for some $s_0<s_1\in\mathbb{R}$. The idea then is to simply trace the curve c from $s=s_0$ to s_1, at which point we have a solution to (5.1).

For the implementation of the algorithm in our case, we let $x_0\in\mathbb{R}^n$ be an initial guess and set $F_0=F(x_0)$. Define $G(x)=F(x)-F_0$ so that the homotopy function is given by

$$H(x,\lambda)=(1-\lambda)G(x)+\lambda F(x)=F(x)+(\lambda-1)F_0.$$

A finite difference approximation to the Jacobian is used. The condition number of the Jacobian has been generally found to be of order 10. A much larger condition number generally indicates an incorrect formulation of the nonlinear equations leading to nonconvergence or convergence to an incorrect solution, that is, a solution to an incomplete set of conditions. In such cases, the plot of the grid for the corresponding mapping function is usually obviously incorrect.

6. Numerical examples. We have developed a MATLAB code which is a revision and upgrade of the code used in [15] for the unbounded case and which extends the code to the bounded case [14]. (A simple, highly symmetric example of a bounded map was computed in [14]; however, the full set of equations for the bounded case were not formulated there.) Some `mfiles` from [19] were used, such as a function for automatically calculating the turning angles from input polygon vertices. The evaluation of the Schwarz–Christoffel integrals needed for the nonlinear equations is done using Gauss–Jacobi quadrature, `gaussj` from [19]. As in [15], a fixed number of

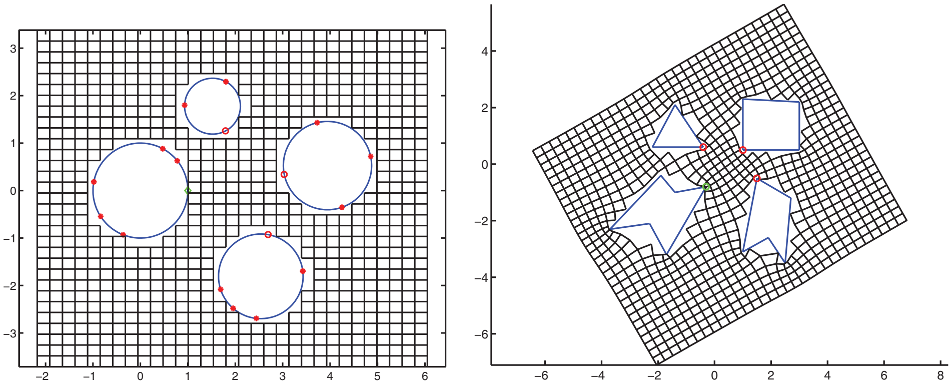

FIG. 6.1. *Example of a map to the exterior of $m = 4$ polygons.*

Gauss–Jacobi points (typically 30) is generally sufficient for each integral. The integrations are along circular arcs, to determine the lengths of the sides of the polygons, and straight line paths between the circles, to determine the relative positions of the circles, as given above. These straight line paths are indicated for our examples in the figures below; see, e.g., Figure 6.2. Since the circles and radii change with each iteration of the nonlinear solver, the integration paths also change. The positions of the circles and the paths can be monitored during the solution. In many cases the circles can move substantially from their initial to their final locations, and the integration paths can often cross nearby circles and prevertices and the iteration still recovers and converges. It was previously believed that such behavior would cause difficult problems and that the integration paths would have to be specified carefully to remain within the computational domain. After numerous experiments with a variety of domains, we have found this to rarely be the case. Just as for the simply connected case [27], the algorithm is extremely robust and usually converges very rapidly, after a few steps of the search method and a final Newton phase, in all but the most extreme domains with no need for accurate initial guesses. The key improvement over previous attempts was the change to unconstrained parameters.

In the following subsections, we give several examples to illustrate these observations. The image of a Cartesian grid in the circle domain is mapped to the polygonal domain in our figures; see Figure 6.1. The final integration paths are plotted, along with separation parameter Δ and its bound $(m-1)^{-1/4}$, which is sufficient for convergence of the infinite products. This bound is often exceeded by Δ with little effect on the algorithm.

6.1. Unbounded domains. In Figure 6.1, we plot a Cartesian grid in the circle domain and its image in the polygonal domain for connectivity $m = 4$. The prevertices on the circles are plotted. The open circles are the "first" (pre)vertices, and the integration paths in between circles are plotted in Figure 6.2. These paths generally connect the "first" vertices or C_1 and the other circles $C_i, i \neq 1$. However, variations are possible, as we'll see below. In this example, $\Delta = 0.6791 < (m-1)^{-1/4} = 0.7598$, so the sufficient condition for convergence of the infinite products is satisfied. In most of our examples $N = 3$ or 4 was sufficient for reasonable accuracy. In Figure 6.2, the $m-1 = 3$ small circles inside of the $m = 4$ large circles are the $N = 1$ level reflections of the large circles in each other. The smallness of the reflected circles gives an

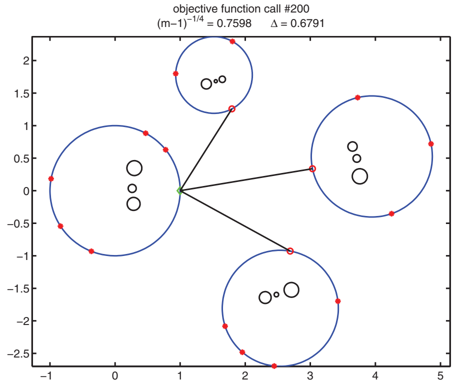

FIG. 6.2. *Objective function integration paths for Figure* 6.1.

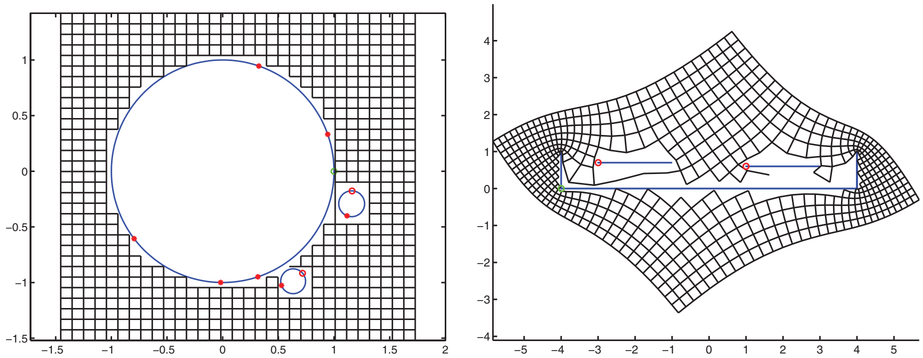

FIG. 6.3. *Slits in a trough.*

illustration of how fast the truncated products approximating $f(z)$ are converging. More careful analysis is given in section 6.4.

The geometry in Figures 6.3 was motivated by the work of van Deursen (see [28, 29]) on printed circuit board design. The thin channels in the trough cause the circles to crowd together in the circle domain. Here, $\Delta = 0.9470 > (m-1)^{-1/4} = 0.8409$, and the sufficient condition for convergence of the infinite products is violated. Such geometries require larger levels of reflection N for sufficient accuracy. Similar domains were calculated in [15, Figs. 5 and 6]. However, it was very difficult to achieve convergence with the old code even for domains with wider channels. Note in Figure 6.4, that one of the final integration paths passes inside the large circle. Since $p_u(z)$ is defined there and no singularities are crossed, this causes no problem.

6.2. Bounded figures. Here we display some examples of maps for bounded domains in Figures 6.5–6.8. Recall that for the bounded case we fix one boundary point $f(z_{1,1}) = w_{1,1}$ and one interior point $f(z_0) = w_0$. All m polygons can then have their positions fixed relative to $f(z_0) = w_0$, as indicated by the integration paths.

6.3. Integration paths for polygon positioning. As originally formulated, the system of nonlinear equations used to solve the parameter problem relies on integrating from the prevertex $z_{1,1}$ to $z_{1,j}$ for $1 < j \leq m$ to position polygons Γ_2 through Γ_m. If care is not taken this can lead to the difficulties shown in Figure 6.9 and

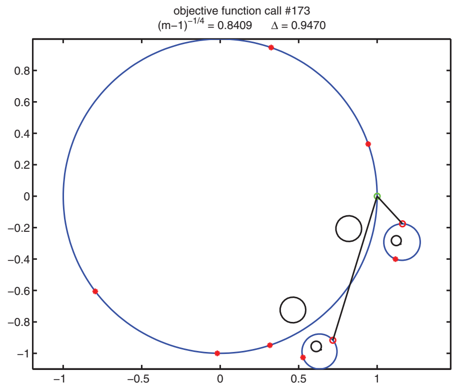

Fig. 6.4. *Objective function integration paths for Figure* 6.3.

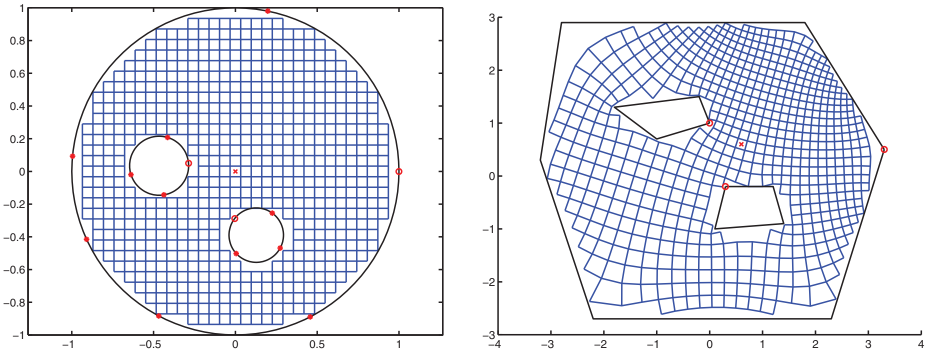

Fig. 6.5. *Simple bounded polygonal example.*

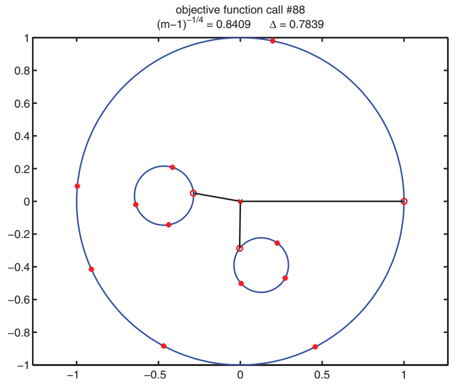

Fig. 6.6. *Objective function integration paths for Figure* 6.5.

Table 6.1. There the integration path used to fix polygon position passes too near the singularities of the Schwarz–Christoffel product—in this case, the reflections of the centers. This problem can be avoided by careful rearrangement of the order of the polygons and the prevertices or by a change of the equations which dictate this posi-

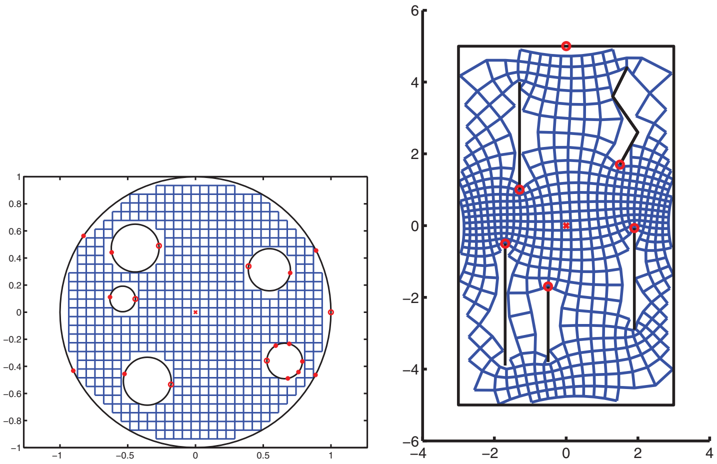

FIG. 6.7. *Multiple slit example, $m = 6$.*

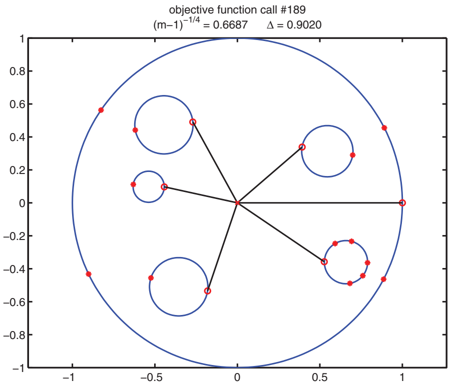

FIG. 6.8. *Objective function integration paths for Figure* 6.7.

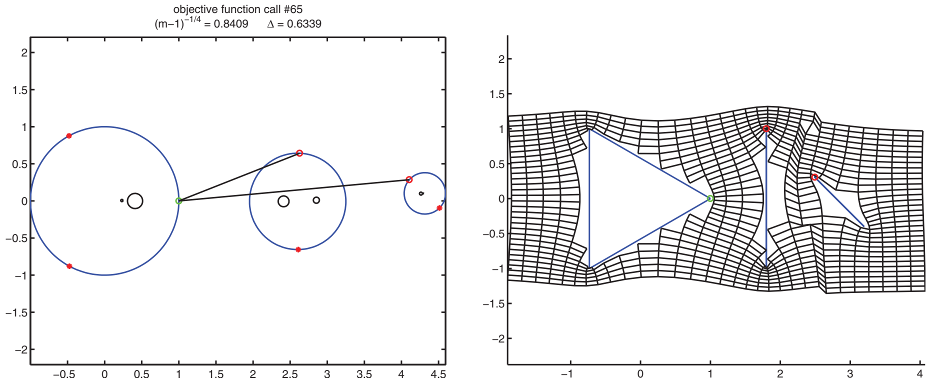

FIG. 6.9. *By anchoring polygons* 2 *and* 3 *to* $z_{1,1}$, *we create an integration path that passes too close to singularities (reflections of the centers) of the Schwarz–Christoffel product. The resulting grid fails to be orthogonal in some places.*

TABLE 6.1

Map parameters found for cases shown in Figures 6.9 *and* 6.10. *It should be noted that continuation was successful in the first case; i.e., the algorithm converged but evidently not to a solution to the parameter problem.*

Figure	c_2	r_2	c_3	r_3
6.9	2.6015 − 0.0061i	0.6492	4.3150 + 0.1017i	0.2784
6.10	2.6042 − 0.0064i	0.6512	4.2119 − 0.0286i	0.2732

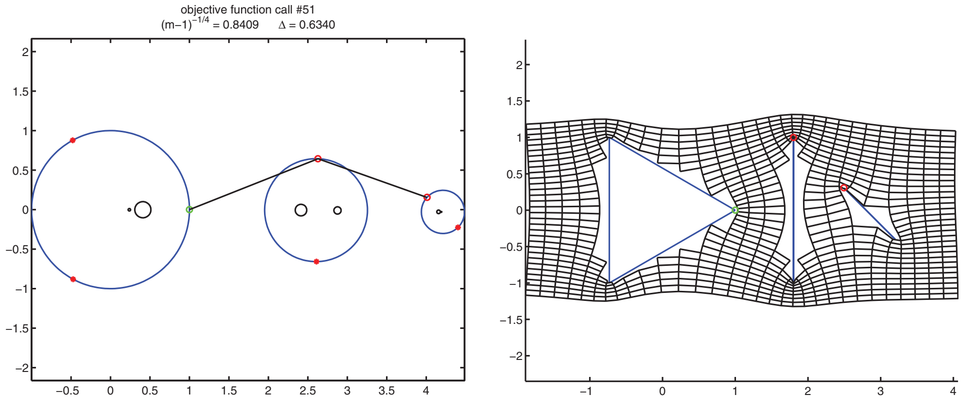

FIG. 6.10. *The order of the polygons is as in the first example, but for positioning we integrate first from* $z_{1,1}$ *to* $z_{1,2}$ *and then from* $z_{1,2}$ *to* $z_{1,3}$.

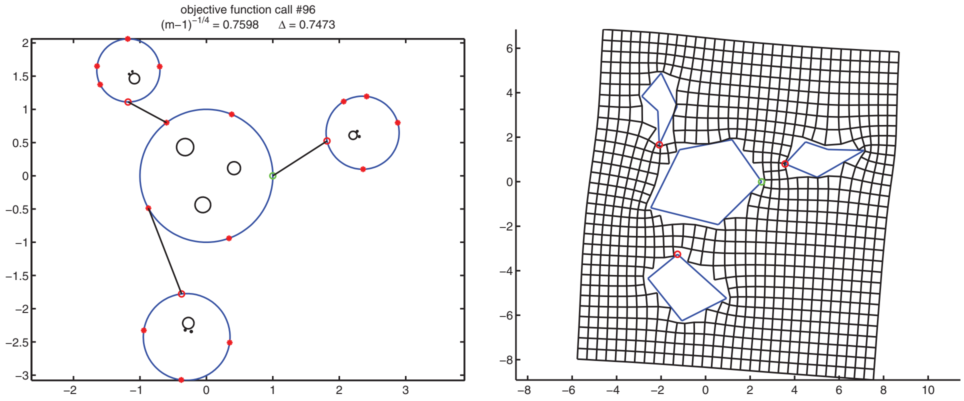
objective function call #96
$(m-1)^{-1/4} = 0.7598$ $\Delta = 0.7473$
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
−3
−2
−1
0
1
2
3
6
4
2
0
−2
−4
−6
−8
−8
−6
−4
−2
0
2
4
6
8
10

FIG. 6.11. *Integration paths for positioning need only connect the first prevertex on the target polygon with another known prevertex, i.e., from* $z_{3,1}$ *to* $z_{1,4}$. *Note that the center polygon is* Γ_1 *with the others, counterclockwise from the bottom, being* Γ_2, Γ_3, *and* Γ_4.

tioning. In the current example, we keep the triangle as Γ_1 and change the equation, which positions the third polygon (the slit on the right) from

$$f(z_{1,3}) - f(z_{1,1}) = w_{1,3} - w_{1,1} \quad \text{to} \quad f(z_{1,3}) - f(z_{1,2}) = w_{1,3} - w_{1,2},$$

using the position of Γ_2 to dictate that of Γ_3. This result is shown in Figure 6.10.

The polygons in Figure 6.11 provide an illustration of a case in which rearranging the order of the polygons and vertices is not likely to produce an integration path

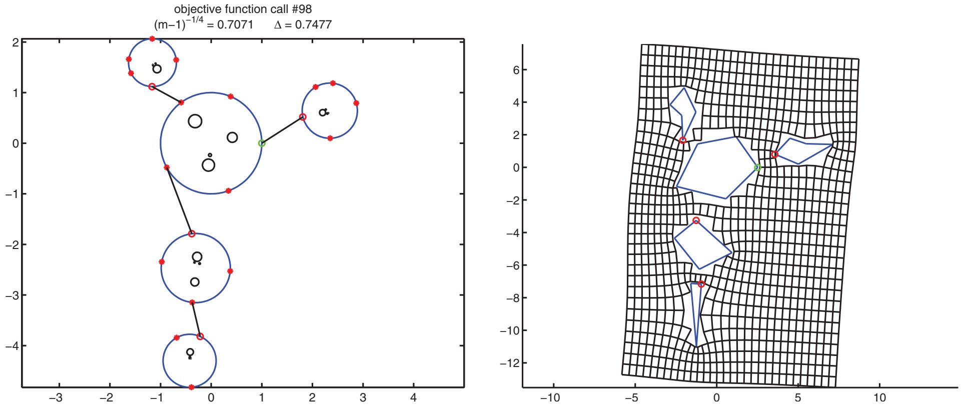

Fig. 6.12. *Integrating from the first polygon and between polygons may also be combined.*

which is completely external to the figures and thus does not come near to possible singularities. For instance, the path from $z_{1,1}$ and $z_{1,4}$ would pass inside of C_1 (Γ_4 is the figure in the upper-left corner). In this case, we integrate from $z_{3,1}$ to $z_{1,4}$ instead and the equations which fix the polygon positions are

$$\begin{aligned} f(z_{1,2}) - f(z_{4,1}) &= w_{1,2} - w_{4,1}, \\ f(z_{1,3}) - f(z_{1,1}) &= w_{1,3} - w_{1,1}, \end{aligned}$$

and

$$f(z_{1,4}) - f(z_{3,1}) = w_{1,4} - w_{3,1}.$$

The requirement that the integration path for positioning be to the first vertex of the polygon being positioned is imposed by the constrained-unconstrained transformation, (4.1) and (4.2); recall that the transformation relies on knowledge of the angle of the first prevertex on a circle to recover the remaining angles. This could likely be changed, but the definition of the first vertex on each polygon is flexible, which effectively removes this concern. As an extension of the previous example, we add a fifth polygon; see Figure 6.12. The addition to the positioning equations above is

$$f(z_{1,5}) - f(z_{3,2}) = w_{1,5} - w_{3,2}.$$

This change to the system of equations for positioning polygons holds for the bounded case as well. In Figure 6.13 we surround a previous example with a rectangular boundary. The system of equations to position the polygons in this case is

$$\begin{aligned} f(z_{1,1}) - f(z_0) &= w_{1,1} - w_0, \\ f(z_{1,2}) - f(z_0) &= w_{1,2} - w_0, \\ f(z_{1,3}) - f(z_{4,2}) &= w_{1,3} - w_{4,2}, \\ f(z_{1,4}) - f(z_0) &= w_{1,4} - w_0, \end{aligned}$$

and

$$f(z_{1,5}) - f(z_{3,2}) = w_{1,5} - w_{3,2}.$$

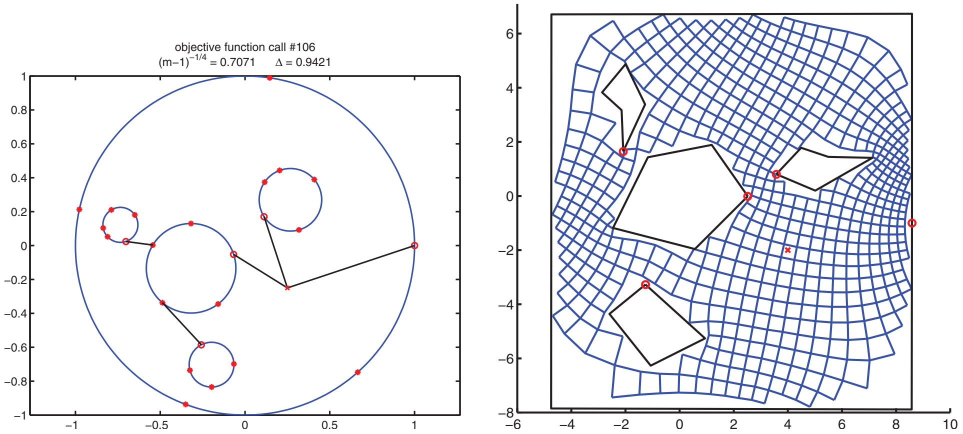

FIG. 6.13. *A choice of integration paths for positioning in the bounded case. (Γ_2 is the largest inner polygon.)*

Note that the map normalization serves as the polygon positioning anchor in both the bounded and unbounded cases. The requirement that $f(z) = A + Bz + O(1/z)$ for $z \to \infty$ in the unbounded domain makes the prevertex $z_{1,1}$ an anchor point. In the bounded domain the requirement that $f(z_0) = w_0$ makes z_0 an anchor. Rewriting the positioning equations requires attention to this condition, as any positioning equation which does not directly or indirectly make use of this information will result in a system for which, in our experience, a search algorithm will fail to find a valid set of map parameters.

6.4. Some error analysis. Here we give a practical method for estimating the error in the mapping function by using the sum of the radii $\sum r_{\nu i}$ for the reflected circles at level $|\nu| = j = N$ as in (2.4) above. Note that the radii $r_{\nu i}$ for the reflected circles can be computed by our reflection routine and, therefore, provide a built-in method for estimating truncation error. We illustrate this error estimate first with the exterior map to three symmetric radial slits displayed in Figure 6.14. The errors and the error estimate are given in Figure 6.15. The errors in the (constrained) parameters (centers, radii, and $\theta_{k,i}$'s at the Nth level denoted by a vector X_N) are measured by $\log \|X_N - X_{N-1}\|_\infty$. The estimate $\log \sum r_{\nu i} - c$ at level N fits the data very well. (In this case, due to the symmetry of the polygonal domain, the $\theta_{k,i}$'s of the prevertices are known exactly.) The radial slit maps used in this example can also be computed "exactly" by infinite product formulas developed in [16]; see also [12] for similar formulas for slit maps given explicitly in terms of Schottky–Klein prime functions. Comparisons of the Schwarz–Christoffel map with the formulas from [16] yield nearly identical errors. Figure 6.16 displays the error and the estimate for the polygonal domain in Figure 6.1.

6.5. Nonlinear solvers. Numerical continuation was chosen for use during the development of the numerics since the method is by design not sensitive to the accuracy of the initial guess. It seems a worthwhile question to ask how other nonlinear equation solvers, specifically those built into MATLAB, handle the parameter problem. Since conformally equivalent domains can have inherently large geometric distortions, it is not clear how one would in general make an initial guess for the map parameters which might be close to a solution. We settle on a reasonable geometry for

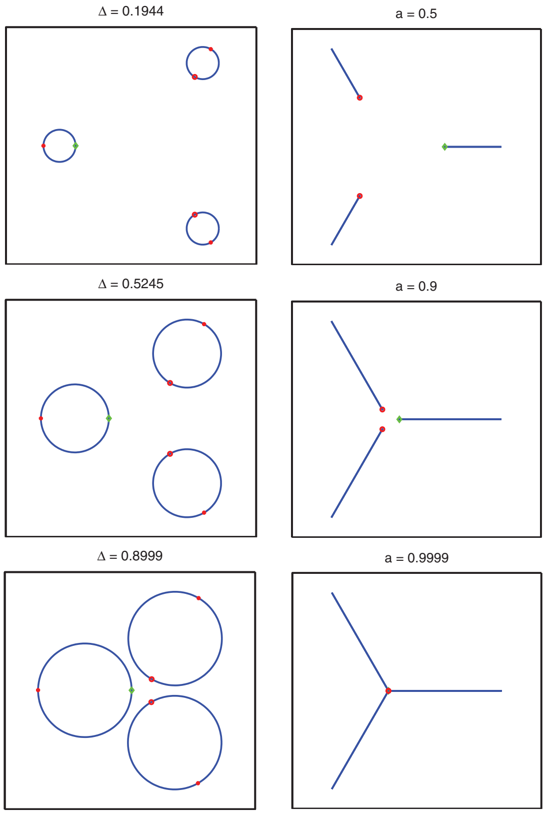

Fig. 6.14. *Geometry used to test numerical accuracy.*

the circle domain for a given polygonal domain. For example, an easy-to-generate initial guess for the polygonal domain and associated circle domain shown in Figure 6.17 is given in Figure 6.18.

We tested the solvers in MATLAB's Optimization Toolbox [25], which could be applied to this problem. These were `fsolve`, `lsqnonlin`, `fminunc`, and `fminsearch`. The `fsolve` function was tested with the trust-region-dogleg (t-r-d), trust-region-reflective (t-r-r), and Levenberg–Marquardt (l-m) algorithms. The function `lsqnonlin` was tested with the (t-r-r) and (l-m) algorithms. Since we did not supply the numerical gradient for our system of equations, we tried the `fminunc` function with the BFGS quasi-Newton algorithm only. Finally `fminsearch` was tried, which uses the Nelder–Mead simplex algorithm. Results of these comparisons are shown in Table 6.2; the infinity norm of the objective function (the system of equations for the parameter problem) and the time in seconds each solver took to find this solution are shown.[1] Note that `fminunc` and `fminsearch` were not tried with connectivity above $m = 3$, since these functions failed to find a solution in the simplest case.

There is also the question of the sensitivity of the MATLAB solvers to the initial guess. As a first attempt to address this question, we make the initial guess shown in Figure 6.19 to solve the problem shown in Figure 6.17. Numerical continuation

[1] The solution time is obviously machine-dependent and is given here only for a sense of the relative speed of each algorithm.

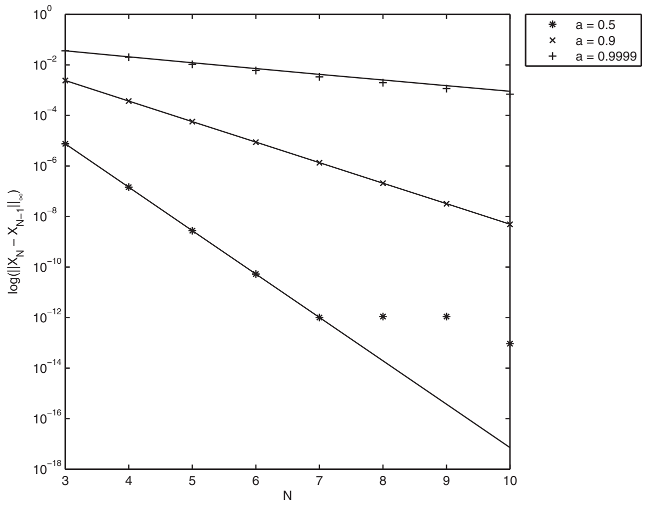

FIG. 6.15. *The log of error vs. levels of reflection N for three examples of maps to the exterior of three symmetric slits; see Figure* 6.14. *The solid-line approximations to the log of the errors are* $\log\sum r_{\nu i} - c$, *where* $c = 0.45, 0.64, 1.26$, *respectively. Note that the case* $a = 0.9999$ *violates the separation condition, since* $\Delta = 0.8999 > (m-1)^{-1/4} = 2^{-1/4} \approx 0.8409$, *and the errors nonetheless decrease, though quite slowly, as one would expect. The errors for the case* $a = .5$ *level off at about* 10^{-12} *due to the tolerance setting in the continuation algorithm, which is set here at* 10^{-15}, *its smallest feasible setting.*

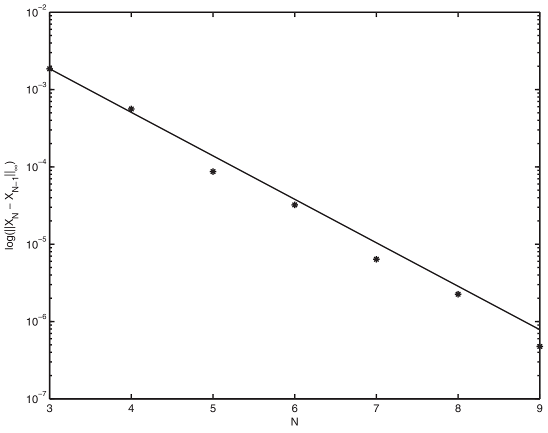

FIG. 6.16. *Error for Figure* 6.1 *approximated with* $\log\sum r_{\nu i} - c$.

handled this new initial guess with no problem, but only the Levenberg–Marquardt algorithm as used in `fsolve` and `lsqnonlin` was able to find a solution; all other attempts failed. The successful results are shown in Table 6.3.

Interestingly, a slight change in the previous poor initial guess (see Figure 6.20) results in a starting point in which the Levenberg–Marquardt algorithm fails for both `fsolve` and `lsqnonlin`. However, using the “trust-region-reflect” algorithm in either of these functions allows a solution to be found; see Table 6.4.

We conclude that, out of the algorithms tried in MATLAB, the numerical continuation algorithm is the most flexible algorithm for the multiply connected parameter problem.

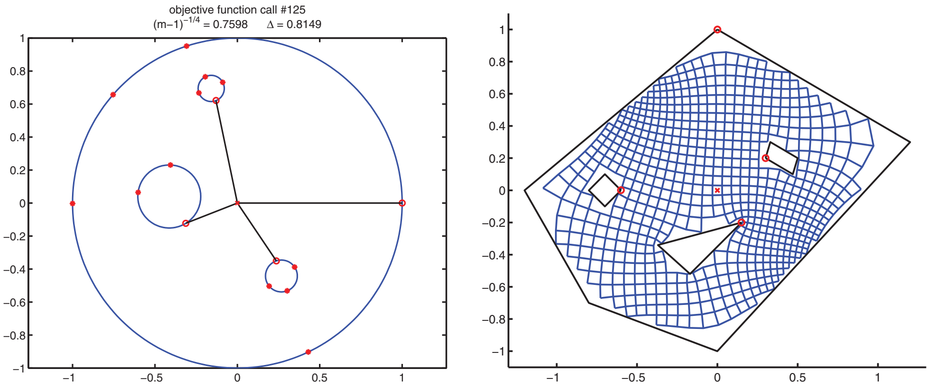

FIG. 6.17. *An example with $m = 4$ used to test the different nonlinear solvers.*

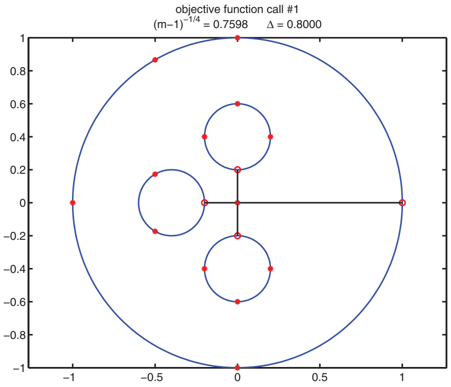

FIG. 6.18. *Initial guess for Figure* 6.17 *used with the nonlinear solvers in Table* 6.2.

TABLE 6.2

Different nonlinear solvers with an initial guess which was not near the solution. The last two solvers in this table were not successful for $m = 3$, and thus were not used for higher values of m. Shown are the infinity norm of the objective function and the time in seconds each solver took to find this solution. Only Figure 6.17 *for $m = 4$ is shown. The other domains just add or delete an inner polygon. Note the increase in timings with m.*

Solver	$m = 3$		$m = 4$		$m = 5$	
	$\|F\|_\infty$	Secs	$\|F\|_\infty$	Secs	$\|F\|_\infty$	Secs
continuation	2.88×10^{-15}	12.74	4.30×10^{-15}	42.63	3.85×10^{-15}	153.97
`fsolve` (t-r-d)	2.32×10^{-10}	12.07	2.49×10^{-10}	57.98	4.63×10^{-12}	278.89
`fsolve` (t-r-r)	1.00×10^{-09}	12.05	2.70×10^{-14}	57.55	4.60×10^{-14}	443.19
`fsolve` (l-m)	4.73×10^{-15}	14.06	2.53×10^{-14}	58.10	4.32×10^{-15}	274.27
`lsqnonlin` (t-r-r)	1.00×10^{-09}	11.84	2.70×10^{-14}	57.36	4.60×10^{-14}	443.44
`lsqnonlin` (l-m)	4.73×10^{-15}	14.21	2.53×10^{-14}	57.53	4.32×10^{-15}	273.25
`fminunc`	3.32×10^{-2}	242.55	-	-	-	-
`fminsearch`	1.58×10^{-1}	266.55	-	-	-	-

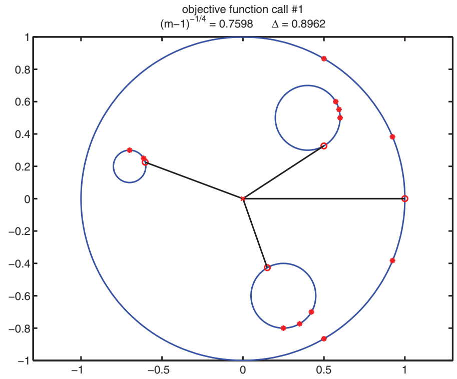

Fig. 6.19. *A poor initial guess.*

Table 6.3

A comparison of the nonlinear solvers which were able to find the map parameters for Figure 6.17, *given the initial guess in Figure* 6.19.

Solver	$\|F\|_\infty$	Secs
continuation	8.160×10^{-15}	135.710
fsolve (l-m)	5.329×10^{-15}	170.573
lsqnonlin (l-m)	5.329×10^{-15}	165.974

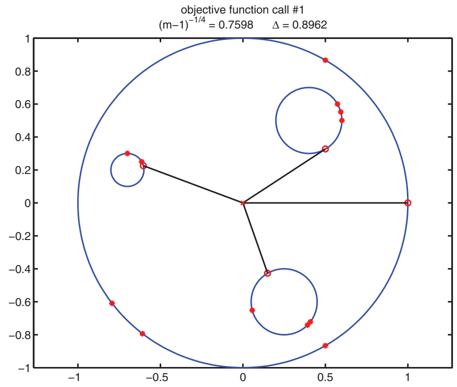

Fig. 6.20. *A slightly modified but still poor initial guess.*

Table 6.4

A comparison of the nonlinear solvers which were able to find the map parameters for Figure 6.17, *given the initial guess in Figure* 6.20.

Solver	$\|F\|_\infty$	Secs
continuation	8.049×10^{-15}	121.571
fsolve (t-r-r)	3.220×10^{-11}	171.911
lsqnonlin (t-r-r)	3.220×10^{-11}	172.406

7. Conclusions and future research. We have developed a numerical method for the computation of Schwarz–Christoffel maps for multiply connected domains. The method is very robust and capable of handling a variety of bounded and unbounded domains. The sufficient conditions guaranteeing the convergence of the infinite products do not seem to be necessary in practice. The formulation in this paper is very effective in most cases of moderate connectivity and sets the stage for numerous applications and future improvements. In this section, we will briefly outline some possible alternative approaches to the numerics, which we hope will lead to further improvements in the accuracy, efficiency, and range of applicability of the formulas. The steady improvements over the last thirty years of methods for the simply connected case can be traced in [20] and subsequent work and should be a reliable guide.

In order to efficiently compute domains with connectivities greater than roughly $m = 5$, the reflection routine will have to be replaced by least squares or Laurent series methods, such as those in [16] for slit maps, for approximating the infinite products in $f'(z)$. There are a few different options for representing $f'(z)$ which we are investigating as bases for such efficient series methods. The paper [18] gives an overview and some indications of how $f'(z)$ can be represented in terms of finite products of slit maps, along with attempts to connect our formulas to those of Crowdy. The Schwarz–Christoffel formulas of Crowdy [7, 8] represent $f'(z)$ as finite products of Schottky–Klein prime functions. Crowdy and Marshall [13] show how to evaluate the Schottky–Klein prime functions using Laurent series, so that convergence of the infinite products is not an issue. (Indeed, a MATLAB code for the prime functions can be downloaded from Crowdy's homepage [11].) Factorizations of $f'(z)$ in terms of slit maps (similar to [18]) using the properties of the Schottky–Klein prime functions directly are discussed in [10] and should prove useful in practice. In addition, the Schottky–Klein prime functions afford a natural tool for solving a number of problems in multiply connected circle domains [9], and combining them numerically with conformal maps, such as the maps developed here or those in [5] for smooth boundaries, will extend their range of applicability to more general domains.

Conformal maps between dissimilar domains can suffer from large geometric distortions, making the problem of computing such maps highly ill-conditioned. The crowding phenomenon, where prevertices at the ends of long channels become close exponentially in the aspect ratio of the channel, is a well-known problem for the simply connected case and has been largely solved; see [3, 20]. In the multiply connected case, elongated channels between boundary components can cause circles to nearly touch; see [5] for examples. If crowding or the location of integration paths places prevertices close to intervals of integration, a careful implementation of adaptive quadrature or of the ideas in [3] will be necessary to maintain accuracy. When circles are close to touching, we expect that the Laurent series will converge slowly and some combination of reflections and Laurent series expansions, as in [6], may be useful.

Domains of very high connectivity will involve very large numbers of unknowns. Here it might be possible to adapt ideas from [4], where the fast multipole method is used to speed up the evaluations of $f'(z)$ and a side-length iteration is used to solve the parameter problem with fewer computations per step than quasi-Newton iterations in order to compute maps of simply connected polygons with thousands of vertices.

Finally, general guidelines or procedures for selecting the integration paths should be developed. For domains of moderate connectivity, allowing the user to select the paths might be best. In addition, an effective procedure for keeping the integration paths inside of the computational domain would be useful in order to avoid rare failures of convergence. Such a procedure is investigated in [24]. We will explore some of these possibilities in future work. We hope to implement these ideas in a publicly available code, such as Schwarz–Christoffel Toolbox [19], with options for user-contributed upgrades and alternatives.

Acknowledgments. The authors thank Toby Driscoll, Alan Elcrat, and John Pfaltzgraff for their interest in and encouragement of this work and for sharing parts of the earlier MATLAB code. They also thank the referees for suggesting a number of improvements and pointing out some additional relevant references.

REFERENCES

[1] E. L. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction*, Springer, New York, 1990.

[2] Y. A. Antipov and A. Y. Zemlyanova, *Motion of a yawed supercavitating wedge beneath a free surface*, SIAM J. Appl. Math., 70 (2009), pp. 923–948.

[3] L. Banjai, *Revisiting the crowding phenomenon in Schwarz–Christoffel mapping*, SIAM J. Sci. Comput., 30 (2008), pp. 618–636.

[4] L. Banjai and L. N. Trefethen, *A multipole method for Schwarz–Christoffel mapping of polygons with thousands of sides*, SIAM J. Sci. Comput., 25 (2003), pp. 1042–1065.

[5] N. Benchama, T. DeLillo, T. Hrycak, and L. Wang, *A simplified Fornberg-like method for the conformal mapping multiply connected regions—Comparisons and crowding*, J. Comput. Appl. Math, 209 (2007), pp. 1–21.

[6] H. Cheng and L. Greengard, *A method of images for the evaluation of electrostatic fields in systems of closely spaced conducting cylinders,* SIAM J. Appl. Math., 58 (1998), pp. 122–141.

[7] D. Crowdy, *The Schwarz-Christoffel mapping to bounded multiply connected polygonal domains,* Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 461 (2005), pp. 2653–2678.

[8] D. Crowdy, *Schwarz-Christoffel mapping to unbounded multiply connected polygonal regions*, Math. Proc. Camb. Phil. Soc., 142 (2007), pp. 319–339.

[9] D. Crowdy, *Geometric function theory: A modern view of a classical subject*, Nonlinearity, 21 (2008), pp. T205–T219.

[10] D. Crowdy, *The Schottky-Klein prime function on the Schottky double of planar domains*, Comput. Methods Funct. Theory, 10 (2010), pp. 501–517.

[11] D. Crowdy, www2.imperial.ac.uk/ dgcrowdy/SKPrime

[12] D. Crowdy and J. Marshall, *Conformal mapping between canonical multiply connected domains*, Comput. Methods Funct. Theory, 6 (2006), pp. 59–76.

[13] D. Crowdy and J. Marshall, *Computing the Schottky-Klein prime function on the Schottky double of planar domains*, Comput. Methods Funct. Theory, 7 (2007), pp. 293–308.

[14] T. K. DeLillo, *Schwarz-Christoffel mapping of bounded, multiply connected domains*, Comput. Methods Funct. Theory, 6 (2006), pp. 275–300.

[15] T. K. DeLillo, T. A. Driscoll, A. R. Elcrat, and J. A. Pfaltzgraff, *Computation of multiply connected Schwarz-Christoffel maps for exterior domains*, Comput. Methods Funct. Theory, 6 (2006), pp. 301–315.

[16] T. K. DeLillo, T. A. Driscoll, A. R. Elcrat, and J. A. Pfaltzgraff, *Radial and circular slit maps of unbounded multiply connected circle domains*, Proc. R. Soc. A., 464 (2008), pp. 1719–1737.

[17] T. K. DeLillo, A. R. Elcrat, and J. A. Pfaltzgraff, *Schwarz-Christoffel mapping of multiply connected domains*, J. d'Anal. Math., 94 (2004), pp. 17–47.

[18] T. DeLillo and E. Kropf, *Slit maps and Schwarz-Christoffel maps for multiply connected domains*, Electron. Trans. Numer. Anal., 36 (2010), pp. 195–223; http://etna.mcs.kent.edu.

[19] T. A. Driscoll, *A MATLAB toolbox for Schwarz-Christoffel mapping*, ACM Trans. Math. Software, 22 (1996), pp. 168–186.

[20] T. A. Driscoll and L. N. Trefethen, *Schwarz-Christoffel Mapping*, Cambridge University Press, Cambridge, UK, 2002.

[21] N. Hale and T. W. Tee, *Conformal maps to multiply slit domains and applications*, SIAM J. Sci. Comput., 31 (2009), pp. 3195–3215.

[22] P. Henrici, *Applied and Computational Complex Analysis*, Vol. III, John Wiley, NY, 1986.

[23] C. Hu, *Algorithm* 785: *A software package for computing Schwarz-Christoffel conformal transformation for doubly connected polygonal regions*, ACM Trans. Math. Software, 24 (1998), pp. 317–333.

[24] M. Landstorfer, *Schwarz-Christoffel Mapping of Multiply Connected Domains–Proof, Discussion, and Elaboration, Matlab Implementation and Search Algorithms for Integration Paths*, Diplomarbeit, Department of Mathematics, Fachhochschule Regensburg, 2007. Available at http://www.mathematik.uni-ulm.de/numerik/staff/mlandsto/.

[25] The MathWorks, Inc., *Optimization Toolbox User's Guide*, 2008.

[26] Z. Nehari, *Conformal Mapping*, Dover, 1952.

[27] L. N. Trefethen, *Numerical computation of the Schwarz–Christoffel transformation*, SIAM J. Sci. Stat. Comput., 1 (1980), pp. 82–102.

[28] A. P. J. van Deursen, *Schwarz-Christoffel analysis of cable conduits with noncontacting cover*, IEEE Electronic Letters, 37, 13 (2001), pp. 816–817.

[29] A. P. J. van Deursen and S. Kapora, *Reduction of inductive common-mode coupling of printed circuits boards by nearby U-shaped metal cabinet panel*, IEEE Trans. on Electromagnetic Compatibility, 47, 3 (2005), pp. 490–497.