SIAM J. SCI. STAT. COMPUT.
Vol. 1, No. 1, March 1980

0196-5204/80/0101-0004 $01.00/0

NUMERICAL COMPUTATION OF THE SCHWARZ–CHRISTOFFEL TRANSFORMATION*

LLOYD N. TREFETHEN†

Abstract. A program is described which computes Schwarz–Christoffel transformations that map the unit disk conformally onto the interior of a bounded or unbounded polygon in the complex plane. The inverse map is also computed. The computational problem is approached by setting up a nonlinear system of equations whose unknowns are essentially the "accessory parameters" z_k. This system is then solved with a packaged subroutine.

New features of this work include the evaluation of integrals within the disk rather than along the boundary, making possible the treatment of unbounded polygons; the use of a compound form of Gauss–Jacobi quadrature to evaluate the Schwarz–Christoffel integral, making possible high accuracy at reasonable cost; and the elimination of constraints in the nonlinear system by a simple change of variables.

Schwarz–Christoffel transformations may be applied to solve the Laplace and Poisson equations and related problems in two-dimensional domains with irregular or unbounded (but not curved or multiply connected) geometries. Computational examples are presented. The time required to solve the mapping problem is roughly proportional to N^3, where N is the number of vertices of the polygon. A typical set of computations to 8-place accuracy with $N \leqq 10$ takes 1 to 10 seconds on an IBM 370/168.

Key words. conformal mapping, Schwarz–Christoffel transformation, Laplace equation, Gauss-Jacobi quadrature

1. Introduction. One of the classical applications of complex analysis is conformal mapping: the mapping of one open region in the complex plane $\mathbb{C}$ onto another by a function which is analytic and one-to-one and has a nonzero derivative everywhere. Such a map preserves angles between intersecting arcs in the domain and image regions; hence the name conformal. The Riemann mapping theorem asserts that any simply connected region in the plane which is not all of $\mathbb{C}$ can be mapped in this way onto any other such region. The problem of constructing such a mapping, however, is in general difficult. We consider here the special case in which the range is the interior of a polygon, where the problem can be considerably simplified.

Suppose that we seek a conformal map from the unit disk in the z-plane to the interior of a polygon P in the w-plane whose vertices are $w_1, \cdots, w_N$, numbered in counterclockwise order. For each k, denote by $\beta_k \pi$ the exterior angle of P at w_k, as indicated in Fig. 1. For any polygon we have a simple relationship among the numbers β_k:

$$\sum_{k=1}^{N} \beta_k = 2. \tag{1.1}$$

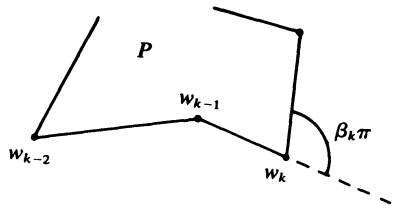


FIG. 1

* Received by the editors May 8, 1979, and in revised form October 9, 1979.

† Computer Science Department, Stanford University, Stanford, California 94305. This work was supported in part by the Office of Naval Research under Contract N00014-75-C-1132, and in part by a National Science Foundation Graduate Fellowship.

If w_k is a finite vertex, we have $-1 \leqq \beta_k < 1$. We shall not require, however, that P be bounded. It may have a number of vertices at complex infinity, and the exterior angles corresponding to these may fall anywhere in the range $1 \leqq \beta_k \leqq 3$. Such angles are defined to be equal to 2π minus the external angle formed in the plane by the intersection of the two sides involved, if they are extended back away from infinity. The example in Fig. 2 should illustrate what is meant by various values of β_k: it is a polygon with five vertices w_k (in this case $w_1 = w_4$), with corresponding values $(\beta_1, \cdots, \beta_5) = (\frac{1}{2}, \frac{4}{3}, \frac{2}{3}, \frac{1}{2}, -1)$. As always, (1.1) holds for this example.

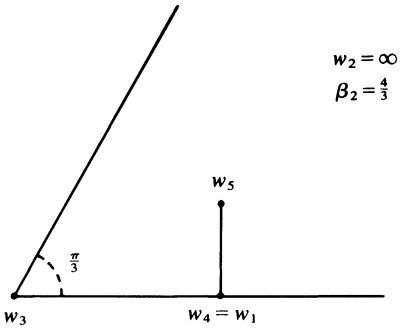


FIG. 2

Let us now pick at random N points z_k ("prevertices") in counterclockwise order around the unit circle and two complex constants C and w_c, and consider the *Schwarz–Christoffel formula*:

$$w = f(z) = w_c + C \int_0^z \prod_{k=1}^{N} \left(1 - \frac{z'}{z_k}\right)^{-\beta_k} dz'. \tag{1.2}$$

The quantities $(1 - z'/z_k)$ always lie in the disk $|w - 1| < 1$ for $|z| < 1$. Therefore, if we choose a branch of $\log(z)$ with a branch cut on the negative real axis by means of which to define the powers in (1.2), $w(z)$ defines an analytic function of z in the disk $|z| < 1$, continuous on $|z| \leqq 1$ except possibly at the vertices z_k.

The Schwarz–Christoffel formula is chosen so as to force the image of the unit disk to have corners in it with the desired exterior angles $\beta_k \pi$. It is not hard to see from (1.2) that at each point z_k, the image $w(z)$ must turn a corner of precisely this angle. This is in keeping with out purpose of mapping the disk onto the interior of P. What the map will in general fail to do is to reproduce the lengths of sides of P correctly, and to be a one-to-one correspondence. Only the angles are guaranteed to come out right.

The variables $z_1, \cdots, z_n$, C, and w_c are the *accessory parameters* of the Schwarz-Christoffel mapping problem. Our first problem—the *parameter problem*—is to determine values of the accessory parameters so that the lengths of sides of the image polygon do come out right. The central theorem of Schwarz–Christoffel transformations asserts that there always exists such a set of accessory parameters.

THEOREM 1 (Schwarz–Christoffel transformation). *Let D be a simply connected region in the complex plane bounded by a polygon P with vertices $z_1, \cdots, z_N$ and exterior angles $\pi\beta_k$, where $-1 \leqq \beta_k < 1$ if z_k is finite and $1 \leqq \beta_k \leqq 3$ if $z_k = \infty$. Then there exists an analytic function mapping the unit disk in the complex plane conformally onto D, and every such function may be written in the form* (1.2).

Proof. The proof is given in [8, Thm. 5.12e].

In fact, for any given polygon there are not just one but infinitely many such conformal mappings. To determine the map uniquely, we may fix exactly three points z_k at will, or fix one point z_k and also fix the complex value w_c, or (as in a standard proof of the Riemann mapping theorem) fix w_c and the argument of the derivative $f'(0)$.

The simplicity of the explicit formula (1.2) is attractive. But because the problem of determining the accessory parameters is intractable analytically, applications of it have almost always been restricted to problems simplified by having very few vertices or one or more axes of symmetry. General Schwarz–Christoffel maps do not appear to have been used as a computational tool, although experiments have been made in computing them.

Problems of numerical conformal mapping have attracted a modest amount of attention for at least thirty years. Gaier [4] produced a comprehensive work describing methods for various problems in this field. For the Schwarz–Christoffel problem, he proposed determining the accessory parameters z_k by setting up a constrained nonlinear system of $N-3$ equations relating (1.2) to the known distances $|w_k - w_j|$, and solving it iteratively by Newton's method [4, p. 171]. Such a procedure has been tried by at least three sets of people; see [7], [10], and [13]. The present work also takes this approach. We believe that this is the first fully practical program for computing Schwarz–Christoffel transformations, however, and the first which is capable of high accuracy without exorbitant cost.

2. Determination of the accessory parameters.

2.1. Formulation as a constrained nonlinear system. The first matter to be settled in formulating the parameter problem numerically is: what parameters in the map (1.2) shall we fix at the outset to determine the Schwarz–Christoffel transformation uniquely? One choice would be to fix three of the boundary points z_k: say, $z_1 = 1$, $z_2 = i$, $z_N = -i$. This normalization has the advantage that the resulting nonlinear system has size only $(N-3)\times(N-3)$, which for a typical problem with $N = 8$ may lead to a solution in less than half the time that a method involving an $(N-1)\times(N-1)$ system requires. Nevertheless, we have chosen here to normalize by the conditions

$$z_N = 1, \qquad w_c = \text{arbitrary point within } P \tag{2.1}$$

which lead to an $(N-1)\times(N-1)$ system. This choice is motivated in part by considerations of numerical scaling: it allows the vertices to distribute themselves more evenly around the unit circle than they might otherwise.(An earlier version of the program mapped from the upper half-plane instead of the unit disk, but was rejected: once points z_k began appearing far from the origin at $x = 10^4$, scaling became a problem.) After a map has been computed according to any normalization, it is of course an easy matter to transform it analytically to a different domain or a different normalization by a Möbius transformation.

Now the nonlinear system must be formulated. The final map must satisfy N complex conditions,

$$w_k - w_c = C\int_0^{z_k} \prod_{j=1}^{N} \left(1-\frac{z'}{z_j}\right)^{-\beta_j} dz', \qquad 1 \leqq k \leqq N. \tag{2.2}$$

These amount to $2N$ real conditions to be satisfied, but they are heavily overdetermined, for the form of the Schwarz–Christoffel formula (1.2) guarantees that the angles will be correct no matter what accessory parameters are chosen. We must reduce the number of operative equations to $N-1$. This is a tricky matter when unbounded polygons are allowed, for one must be careful that enough information about the polygon P is retained that no degrees of freedom remain in the computed solution.

We proceed as follows. First, we require that every connected component of P contain at least one vertex w_k. Thus even an infinite straight boundary must be considered to contain a (degenerate) vertex. This restriction eliminates any translational degrees of freedom. Second, at least one component of P must in fact contain two finite vertices, and w_N and w_1 will be taken to be two such. This restriction eliminates rotational degrees of freedom.

Now define

$$C = (w_N - w_c) \Big/ \int_0^{z_N} \prod_{j=1}^{N} \left(1 - \frac{z'}{z_j}\right)^{-\beta_j} dz', \tag{2.3}$$

where $z_N = 1$ is fixed permanently by (2.1). Next, impose the complex condition (real equations 1, 2)

$$w_1 - w_c = C \int_0^{z_1} \prod_{j=1}^{N} \left(1 - \frac{z'}{z_j}\right)^{-\beta_j} dz'. \tag{2.4a}$$

This amounts to two real equations to be satisfied.

Denote by $\Gamma_1, \cdots, \Gamma_m$ the distinct connected components of P, numbered in counterclockwise order. For each $l \geqq 2$, impose one more complex condition: if z_{k_l} is the last vertex of Γ_l in the counterclockwise direction, then (real equations $3, 4, \cdots, 2m$)

$$w_{k_l} - w_c = C \int_0^{z_{k_l}} \prod_{j=1}^{N} \left(1 - \frac{z'}{z_j}\right)^{-\beta_j} dz'. \tag{2.4b}$$

Finally, $N - 2m - 1$ conditions of side length are imposed. For each pair (z_k, z_{k+1}) beginning at $k = 1$ and moving counterclockwise, where both vertices are finite, we require (real equations $2m+1, \cdots, N-1$)

$$|w_{k+1} - w_k| = \left| C \int_{z_k}^{z_{k+1}} \prod_{j=1}^{N} \left(1 - \frac{z'}{z_j}\right)^{-\beta_j} dz' \right| \tag{2.4c}$$

until a total of $N-1$ conditions have been imposed. If P contains at least one vertex at infinity, then every bounded side will have been represented in a condition of the form (2.4c) except for the side (w_N, w_1), which is already taken care of by (2.1) and (2.4a). If P is bounded, then the last two sides in counterclockwise order—(w_{N-2}, w_{N-1}) and (w_{N-1}, w_N)—will not be so represented.

We have not stated over what contours the integrals of (2.4) are defined. This does not matter mathematically, as the integrand is analytic, but it may matter numerically. In this work we have evaluated them always over the straight line segment between the two endpoints, a procedure which poses no domain problems since the unit disk is strictly convex. Figure 3 illustrates what contours are involved in computing the integrals in (2.3) and (2.4) for a sample case with $N = 10$, $m = 3$.

The nonlinear system is now determined, and its unique solution will give the unknown parameters C and $z_1, \cdots, z_{N-1}$ for the Schwarz–Christoffel mapping. We must, however, take note of two special cases in which the solution is not completely determined by (2.4). It was remarked that if P is bounded, then nowhere in (2.4) does the point w_{N-1} appear. If $\beta_{N-1} \neq -1$ or 0, then this omission is of no consequence for the geometry of the problem forces w_{N-1} to be correct. If $\beta_{N-1} = 0$ or -1, however, then w_{N-1} is not determined *a priori*. The former case is of little consequence, for since $\beta_{N-1} = 0$, the value taken for z_{N-1} has no effect on the computed mapping, as may be seen in (1.2), nor is there any purpose in including w_{N-1} among the vertices of P in the

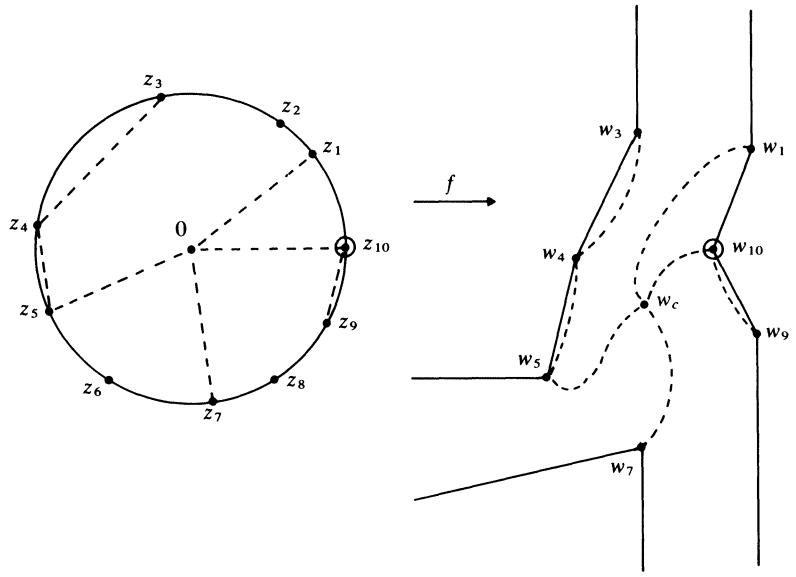


FIG. 3. *Contours of integration within the disk. A sample Schwarz–Christoffel problem is shown with* $N = 10$ *vertices of which* $m = 3$ *vertices are at infinity, illustrating what integrals are computed to evaluate the system* (2.4): (i) 1 *radial integral along* $(0 - z_{10})$ *defines* C (2.3); (ii) 1 *radial integral along* $(0 - z_1)$ *determines two real equations to fix* w_1 (2.4a); (iii) 2 *radial integrals along* $(0 - z_5)$ *and* $(0 - z_7)$ *determine four real equations to fix* w_5 *and* w_7 (2.4b); (iv) 3 *chordal integrals along* $(z_3 - z_4)$, $(z_4 - z_5)$, *and* $(z_9 - z_{10})$ *determine three real equations to fix* $|w_4 - w_3|$, $|w_5 - w_4|$, *and* $|w_{10} - w_9|$ (2.4c). *TOTAL:* $N - 1 = 9$ *real equations.*

first place. (Still, there may be problems in solving the system (2.4) numerically, for it is now undetermined.) The latter case, $\beta_{N-1} = -1$, is more serious, and must be avoided in the numbering of the vertices w_k.

2.2. Transformation to an unconstrained system. The nonlinear system (2.4) ostensibly involves $N-1$ complex unknown points $z_1, \cdots, z_{N-1}$ on the unit circle. In dealing with such a system, we naturally begin by considering not the points z_k themselves, but their arguments θ_k given by

$$z_k = e^{i\theta_k}, \qquad 0 < \theta_k \leqq 2\pi. \tag{2.5}$$

Now the system depends on $N-1$ real unknowns, and the solution in terms of the θ_k is fully determined.

However, the system (2.4) as it stands must be subject to a set of strict inequality constraints,

$$0 < \theta_k < \theta_{k+1}, \qquad 1 \leqq k \leqq N-1, \tag{2.6}$$

which embody the fact that the vertices z_k must lie in ascending order counterclockwise around the unit circle. To solve the system numerically, it is desirable to eliminate these constraints somehow. We do this by transforming (2.4) to a system in $N-1$ variables $y_1, \cdots, y_{N-1}$, defined by the formula

$$y_k = \log \frac{\theta_k - \theta_{k-1}}{\theta_{k+1} - \theta_k}, \qquad 1 \leqq k \leqq N-1, \tag{2.7}$$

where θ_0 and θ_N, two different names for the argument of $z_N = 1$, are taken for convenience as 0 and 2π, respectively.

We make no attempt to tailor the numerical solution procedure to the particular Schwarz–Christoffel problem under consideration. In particular, all iterations begin with the trivial initial estimate $y_k = 0$ $(1 \leqq k \leqq N-1)$. This corresponds to trial vertices spaced evenly around the unit circle. The following input parameters to NS01A have generally remained fixed: DSTEP $= 10^{-8}$ (step size used to estimate derivatives by finite differences), DMAX $= 10$ (maximum step size), MAXFUN $= 15(N-1)$ (maximum number of iterations).

A fourth parameter, EPS, defines the convergence criterion—how large a function vector (square root of sum of squares of functions values) will be considered to be satisfactorily close to zero. We have most often taken 10^{-8} or 10^{-14} here. The choice of EPS is not very critical, however, as convergence in NS01A is generally quite fast in the later stages.

In the course of this work about two hundred Schwarz–Christoffel transformations have been computed, ranging in complexity from $N = 3$ to $N = 20$. NS01A has converged successfully to an accurate solution in nearly all of these trials. Section 5.1 gives a series of plots showing this convergence graphically for a simple example.

3. Computation of the S–C map and its inverse. Determining the accessory parameters is the most formidable task in computing numerical Schwarz–Christoffel transformations. Once this is done, evaluation of the map and of its inverse follow relatively easily. The foundation of these computations continues to be compound Gauss–Jacobi quadrature.

3.1. From disk to polygon: $w = w(z)$. To evaluate the forward map $w(z)$ for a given point z in the disk or on the circle, we must compute the integral

$$w = w_0 + C\int_{z_0}^{z} \prod_{j=1}^{N} \left(1 - \frac{z'}{z_j}\right)^{-\beta_j} dz' \tag{3.1}$$

with $w_0 = w(z_0)$, where the endpoint z_0 may be any point in the closed disk at which the image $w(z_0)$ is known and not infinite. Three possible choices for z_0 suggest themselves:

(1) $z_0 = 0$; hence $w_0 = w_c$;

(2) $z_0 = z_k$ for some k; hence $w_0 = w_k$, a vertex of P;

(3) $z_0 =$ some other point in the disk at which w has previously been computed.

In cases (1) and (3), neither endpoint has a singularity, and an evaluation of (3.1) by compound Gauss–Jacobi quadrature reduces to the use of compound Gauss quadrature. In case (2) a singularity of the form $(1 - z/z_k)^{-\beta_k}$ is present at one of the endpoints and the other endpoint has no singularity.

The best rule for computing $w(z)$ is: if z is close to a singular point z_k (but not one with $w_k = \infty$), use choice (2); otherwise, use choice (1). In either case we employ compound Gauss–Jacobi quadrature, taking normally the same number of nodes as was used in solving the parameter problem. By this procedure we evaluate $w(z)$ readily to "full" accuracy—that is, the accuracy to which the accessory parameters have been computed, which is directly related to the number of points chosen for Gauss–Jacobi quadrature (see § 4.1). Quadrature nodes and weights need only be computed once, of course.

We should emphasize that even in the vicinity of a singularity z_k, the evaluation of the map $w = w(z)$ is inherently very accurate. This very satisfactory treatment of singular vertices is a considerable attraction of the Schwarz–Christoffel approach for solving problems of Laplace type. In particular, in a potential problem the Schwarz–Christoffel transformation "automatically" handles the singularities correctly at any number of reentrant corners.

3.2. From polygon to disk: $z = z(w)$. For computing the inverse mapping $z = z(w)$, at least two possibilities exist, both of them quite powerful. The most straightforward approach is to view the formula $w(z) = w$ as a nonlinear equation to be solved for z, given some fixed value w. The solution may then be found iteratively by Newton's method or a related device. $w(z)$ should be evaluated at each step of such a process by compound Gauss–Jacobi quadrature along a straight line segment whose initial point remains fixed throughout the iteration.

An alternative approach is to invert the Schwarz–Christoffel formula

$$\frac{dw}{dz} = C \prod_{k=1}^{N} \left(1 - \frac{z}{z_k}\right)^{-\beta_k}$$

to yield the formula

$$\frac{dz}{dw} = \frac{1}{C} \prod_{k=1}^{N} \left(1 - \frac{z}{z_k}\right)^{+\beta_k}. \tag{3.2}$$

This inversion is possible because $w = w(z)$ is a conformal mapping, which means $|dw/dz| > 0$ everywhere. Equation (3.2) may now be thought of as an ordinary differential equation (o.d.e.),

$$\frac{dz}{dw} = g(w, z), \tag{3.3}$$

in one complex variable w. If a pair of values (z_0, w_0) is known and the new value $z = z(w)$ is sought, then z may be computed by applying a numerical o.d.e. solver to the problem (3.3), taking as a path of integration any curve from w_0 to w which lies within the polygon P.

In our program we have chosen to combine these two methods, using the second method to generate an initial estimate for use in the first. We begin with the o.d.e. formulation, using the code ODE by Shampine and Gordon, and for convenience we integrate whenever possible along the straight line segment from w_c to w. (ODE, like most o.d.e. codes, is written for problems in real arithmetic, so that we must first express (3.2) as a system of first-order o.d.e.'s in two real variables.) Since P may not be convex, more than one line segment step may be required to get from w_0 to w in this way. It will not do to take $w_0 = w_k$ for some vertex w_k without special care, because (3.2) is singular at w_k.

From ODE we get a rough estimate $\tilde{z}$ of $z(w)$, accurate to roughly 10^{-2}. This estimate is now used as an initial guess in a Newton iteration to solve the equation $w(z) = w$. This method is faster than the o.d.e. formulation for getting a high-accuracy answer. More important, it is based on the central Gauss–Jacobi quadrature routine unlike the o.d.e. computation.

In summary, we compute the inverse map $z = z(w)$ rapidly to full accuracy by the following steps:

(1) Solve (3.2) to low accuracy with a packaged o.d.e. solver, integrating whenever possible along the line segment from w_c to w; call the result $\tilde{z}$;
(2) Solve the equation $w(z) = w$ for z by Newton's method, using $\tilde{z}$ as an initial guess.

4. Accuracy and speed.

4.1. Accuracy. The central computational step is the evaluation of the Schwarz–Christoffel integral, and the accuracy of this evaluation normally determines the accuracy of the overall computation. As a consequence of the quadrature principle

At each iterative step in the solution of the nonlinear system (2.4), we begin by computing a set of angles $\{\theta_k\}$ and then vertices $\{z_k\}$ from the current trial set $\{y_k\}$. This is easy to do, though not immediate since (2.7) are coupled. In this way the problem is reduced to one of solving an unconstrained nonlinear system of equations in $N-1$ real variables.

2.3. Integration by compound Gauss–Jacobi quadrature. The central computation in solving the parameter problem, and indeed in all Schwarz–Christoffel computations, is the numerical evaluation of the Schwarz–Christoffel integral (1.2) along some path of integration. Typically one or both endpoints of this path are prevertices z_k on the unit circle, and in this case a singularity of the form $(1-z/z_k)^{-\beta_k}$ is present in the integrand at one or both endpoints.

A natural way to compute such integrals quickly is by means of Gauss–Jacobi quadrature (see [2, p. 75]). A Gauss–Jacobi quadrature formula is a sum $\sum_{i=1}^{N_{\text{quad}}} w_i f(x_i)$, where the weights w_i and nodes x_i have been chosen in such a way that the formula computes the integral $\int_{-1}^{+1} f(x)(1-x)^{\alpha}(1+x)^{\beta}\,dx$ exactly for $f(x)$ a polynomial of as high a degree as possible. Thus Gauss–Jacobi quadrature is a generalization of pure Gaussian quadrature to the case where singularities of the general form $(1-x)^{\alpha}(1+x)^{\beta}$ $(a, \beta > -1)$ are present. The required nodes and weights can be computed numerically; we have used the program GAUSSQ by Golub and Welsch [5] for this purpose.

Gauss–Jacobi quadrature appears made-to-order for the Schwarz–Christoffel problem, and at least three previous experimenters have used it or a closely related technique [7], [10], [13]. We began by doing the same, and got good results for many polygons with a small number of vertices. In general, however, we found this method of integration very inaccurate. For a typical sample problem with $N = 12$ and $N_{\text{quad}} = 8$, it produced integrals accurate to only about 10^{-2}, and it does much worse if one chooses polygons designed to be troublesome.

What goes wrong is a matter of resolution. Consider a problem like the one shown in Fig. 4. We wish to compute the integral (1.2) along the segment from z_k to some point p. (In the parameter problem p might be 0 or z_{k-1}; in later computations it might be any point in the disk.) Now direct application of a Gauss–Jacobi formula will involve sampling the integrand at only N_{quad} nodes between z_k and p. If the singularity z_{k+1} is so close to the path of integration that the distance $\varepsilon = |z_{k+1} - z_k|$ is comparable to the distance between nodes, then obviously the Gauss–Jacobi formula will yield a very poor result. It turns out that in Schwarz–Christoffel problems the correct spacing of prevertices z_k around the unit circle is typically very irregular, so the appearance of this problem of resolution is the rule, not the exception. (See examples in § 5.)

To maintain high accuracy without giving up much speed, we have switched to a kind of *compound Gauss–Jacobi quadrature* (see [2, p. 56]). We adopt, somewhat arbitrarily, the following quadrature principle:

> No singularity z_k shall lie closer to an interval of integration than half the length of that interval.

To achieve this goal, our quadrature subroutine must be able to divide an interval of integration into shorter subintervals as necessary, working from the endpoints in. On the short subinterval adjacent to the endpoint, Gauss–Jacobi quadrature will be applied; on the longer interval (or intervals) away from the endpoint, pure Gaussian quadrature will be applied. The effect of this procedure is that number of integrand evaluations required to achieve a given accuracy is reduced from $O(1/\varepsilon)$ to $O(\log_2 1/\varepsilon)$.

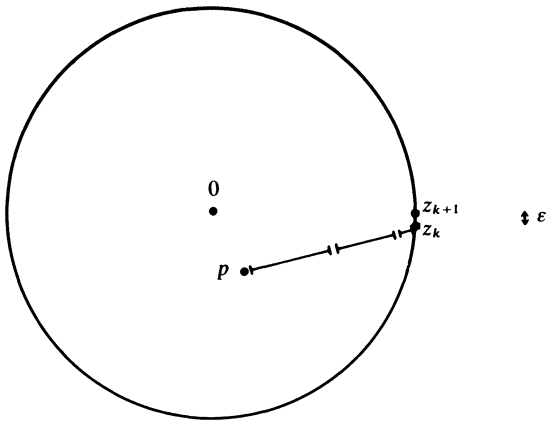


FIG. 4. *Compound Gauss–Jacobi quadrature. Division of an interval of integration into subintervals to maintain desired resolution.*

Figure 4 shows the intervals of integration that come into play in compound Gauss–Jacobi quadrature. For a plot comparing the accuracy of simple and compound Gauss–Jacobi quadrature in another typical problem see § 4.1.

With the use of compound Gauss–Jacobi quadrature, we now achieve high accuracy in little more than the time that direct Gauss–Jacobi quadrature takes. This is possible because only a minority of integrals have a singularity close enough that subdivision of the interval of integration is required. In the 12-vertex example mentioned above, the switch to compound Gauss–Jacobi integration decreased the error from 10^{-2} to $2 \cdot 10^{-7}$.

There remains one circumstance in which integration by compound Gauss–Jacobi quadrature as described here is unsuccessful. This is the case of an integration interval with one endpoint quite near to some prevertex z_k corresponding to a vertex $w_k = \infty$. We cannot evaluate such an integral by considering an interval which begins at z_k, for the integral would then be infinite. The proper approach to this problem is probably the use of integration by parts, which can reduce the singular integrand to one that is not infinite. Depending on the angle β_k, one to three applications of integration by parts will be needed to achieve this. We have not implemented this procedure.

The subtlety of the integration problem in Schwarz–Christoffel computations is worth emphasizing. It is customary to dispatch the integration problem as quickly as possible, in order to concentrate on the "difficult" questions: computation of accessory parameters and inversion of the Schwarz–Christoffel map. We believe, however, that the more primary problem of computing Schwarz–Christoffel integrals—the "forward" problem—should always remain a central concern. Any numerical approach to the parameter problem or the inversion problem is likely to employ an iterative scheme which depends at each step on an evaluation of the integral (1.2), and so the results can only be as accurate as that evaluation.

2.4. Solution of system by packaged solver. The unconstrained nonlinear system is now in place and ready to be solved. For this purpose we employ a library subroutine: NS01A, by M. J. D. Powell [11], which uses a steepest descent search in early iterations if necessary, followed by a variant of Newton's method later on. (The routine does not use analytic derivatives.) It is assumed that a variety of other routines would have served comparably well.

adopted in § 2.3—that no quadrature interval shall be longer than twice the distance to the nearest singularity z_k—the compound Gauss–Jacobi formulation achieves essentially the full accuracy typical of Gaussian quadrature rules operating upon smooth integrands. That is, the number of digits of accuracy is closely proportional to N_{quad}, the number of quadrature nodes per half-interval, with a very satisfactory porportionality constant in practice of approximately 1.

It is important not only to be capable of high accuracy, but to be able to measure how much accuracy one has in fact achieved in a given computation. To do this we employ an accuracy testing subroutine, which is regularly called immediately after the parameter problem is solved. Given a computed set of accessory parameters C and $\{z_k\}$,

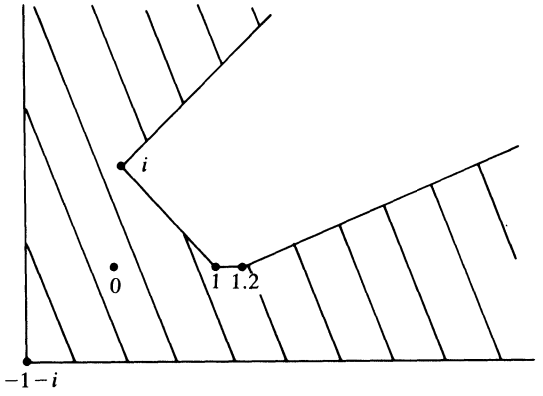


FIG. 5(a)

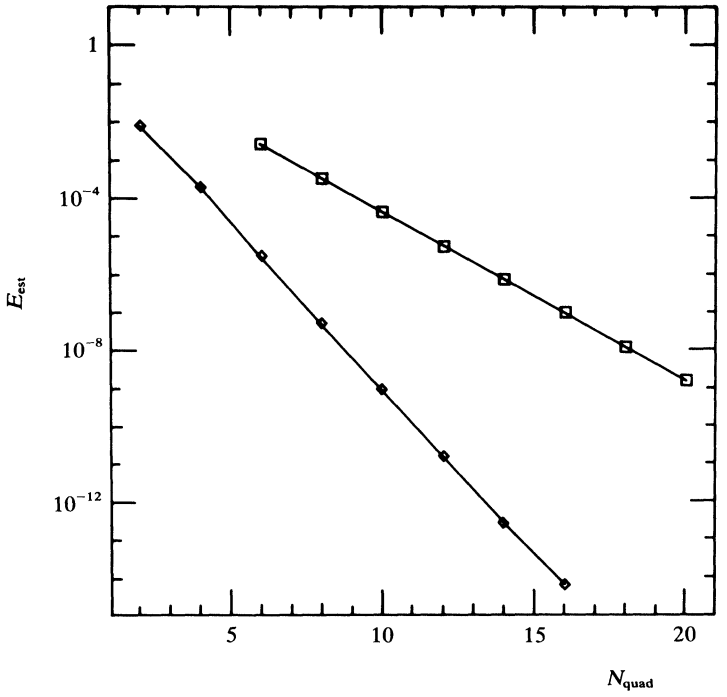


FIG. 5(b). *Quadrature accuracy as a function of number of nodes. The error estimate E_{est} is plotted as a function of N_{quad} for the polygon of Fig.* 5(a). *The upper and lower curves correspond to simple Gauss–Jacobi and compound Gauss–Jacobi quadrature, respectively.*

this subroutine computes the distances $|w_k - w_c|$ for each $w_k \neq \infty$ and the distances $|w_{k-1} - w_{k+1}|$ for each $w_k = \infty$, making use of the standard routine for compound Gauss–Jacobi quadrature. The numbers obtained are compared with the exact distances specified by the geometry of the polygon, and the maximum error, E_{est}, is printed as an indication of the magnitude of errors in the converged solution. It is now probable that subsequent computations of $w(z)$ or $z(w)$ will have errors no greater than roughly E_{est}.

Most often we have chosen to use an 8-point quadrature formula. Since each interval of integration is initially divided in half by the quadrature subroutine, this means in reality at least 16 nodes per integration. With this choice E_{est} consistently has magnitude $\sim 10^{-8}$ for polygons on the scale of unity.

Figure 5b gives an indication of the relationship between the number of quadrature nodes and the error E_{est}; it shows E_{est} as a function of N_{quad} for a 6-gon which is shown in Fig. 5a. Two curves are shown: one for simple Gauss–Jacobi quadrature, and one for compound Gauss–Jacobi quadrature. The exact quantities here should not be taken too seriously; examples could easily have been devised to make the difference in performance of the two quadrature methods much smaller or much greater.

4.2. Speed. Any application of Schwarz–Christoffel transformations consists of a sequence of steps; for convenience we use the names of the corresponding subroutines in our program:

INIT—set up problem
QINIT—compute quadrature nodes and weights
SCSOLV—solve parameter problem
TEST—estimate accuracy of solution
ZSC, WSC, etc.—compute forward and inverse transformations in various applications

Among these tasks INIT, QINIT and TEST all take negligible amounts of time relative to the other computations: typically less than 0.1 secs. on the IBM 370/168 for INIT and QINIT, and for TEST a variable time that is usually less than 5% of the time required by SCSOLV. What remains are three main time consumers: SCSOLV, ZSC, and WSC.

We begin with WSC, which performs the central evaluation of (1.2) by compound Gauss–Jacobi quadrature. This evaluation takes time proportional to N_{quad} (the number of quadrature nodes) and to N (the number of vertices). The first proportionality is obvious, and the second results from the fact that the integrand of (1.2) is an N-fold product. Very roughly, we may estimate

$$\text{time to solve } w = w(z)\text{: } 0.25 \cdot N_{\text{quad}} \cdot N \text{ msec.} \tag{4.1a}$$

for double precision computations on the IBM 370/168. Taking a typical value of $N_{\text{quad}} = 8$, which normally leads to 8-digit accuracy, (4.1a) may be rewritten

$$\text{time to solve } w = w(z)\text{: } 2N \text{ msec.} \tag{4.1b}$$

For the minority of cases in which the interval must be subdivided to maintain the required resolution, these figures will be larger.

To estimate the time required to solve the parameter problem, we combine (4.1) with an estimate of how many integrals must be computed in the course of solving this problem. To begin with, at each iteration about N integrals are required by NS01A (the exact number depends on the number of vertices at infinity). On top of this, it is a fair

estimate to say that $4N$ iterations will be required by NS01A to achieve a high-accuracy solution. We are therefore led to the estimate

(4.2a) time to solve parameter problem: $N_{\text{quad}} \cdot N^3$ msec.

or, taking again $N_{\text{quad}} = 8$,

(4.2b) time to solve parameter problem: $8N^3$ msec.

These estimates correspond fairly well with observed computation times for the parameter problem: two problems with $N = 5$ and $N = 18$ may be expected to take about 1 and 50 seconds, respectively. It is clear that computing a Schwarz–Christoffel transformation becomes quite a sizeable problem for polygons with more than ten vertices. In particular, such computations are too time-consuming for it to be very practical to approximate a curved domain by a polygon with a large number of vertices.

Finally, we must consider the time taken by subroutine ZSC to invert the Schwarz–Christoffel map. This too is proportional to N_{quad}, and quite problem dependent. We estimate very roughly:

(4.3a) time to solve $z = z(w)$: $N_{\text{quad}} \cdot N$ msec.

or, with $N_{\text{quad}} = 8$,

(4.3b) time to solve $z = z(w)$: $8N$ msec.

Note that inverting the Schwarz–Christoffel map is only about four times as time-consuming as computing it in the forward direction.

In practice, computational applications will vary considerably in the use they make of a Schwarz–Christoffel transformation once the parameter problem is solved. If only a few dozen applications of ZSC or WSC are required, then the computational time for solving the parameter problem will dominate. If thousands of such computations are needed, on the other hand, then the parameter problem may become relatively insignificant. The latter situation is most likely to hold when plotting is being done, or when a high-accuracy solution in the model domain is to be computed by means of finite differences.

In summary, high accuracy is cheap in Schwarz–Christoffel transformations; what consumes time is solving problems involving a large number of vertices.

5. Computed examples and applications.

5.1. Iterative process for a single example. Figure 6 shows graphically the process of convergence from the initial estimate in an example involving a 4-gon. Routine NS01A begins by evaluating the function vector (2.4) at the initial guess, then at each of $N-1$ input vectors determined by perturbing the initial guess by the small quantity DSTEP in each component. As a result, the first N pictures always look almost alike, which is why the series shown begins at IT = 4 rather than IT = 1. Each plot shows the current image polygon together with the images of concentric circles in the unit disk (which appear as "contours") and the images of radii leading from the center of the disk to the current prevertices z_k.

These pictures have an elegant bonus feature about them: they may be interpreted as showing not only the image polygon but simultaneously the domain disk, including the prevertices z_k along the unit circle. To see this, look at one of the inner "contour" curves, one which is apparently circular, and the radii within it. Since $w = w(z)$ is a conformal map within the interior of the disk, the radii visible in this circle must intersect at the same angles as their preimages in the domain disk. Thus the inner part of any one of these image plots is a faithful representation on a small scale of the circular

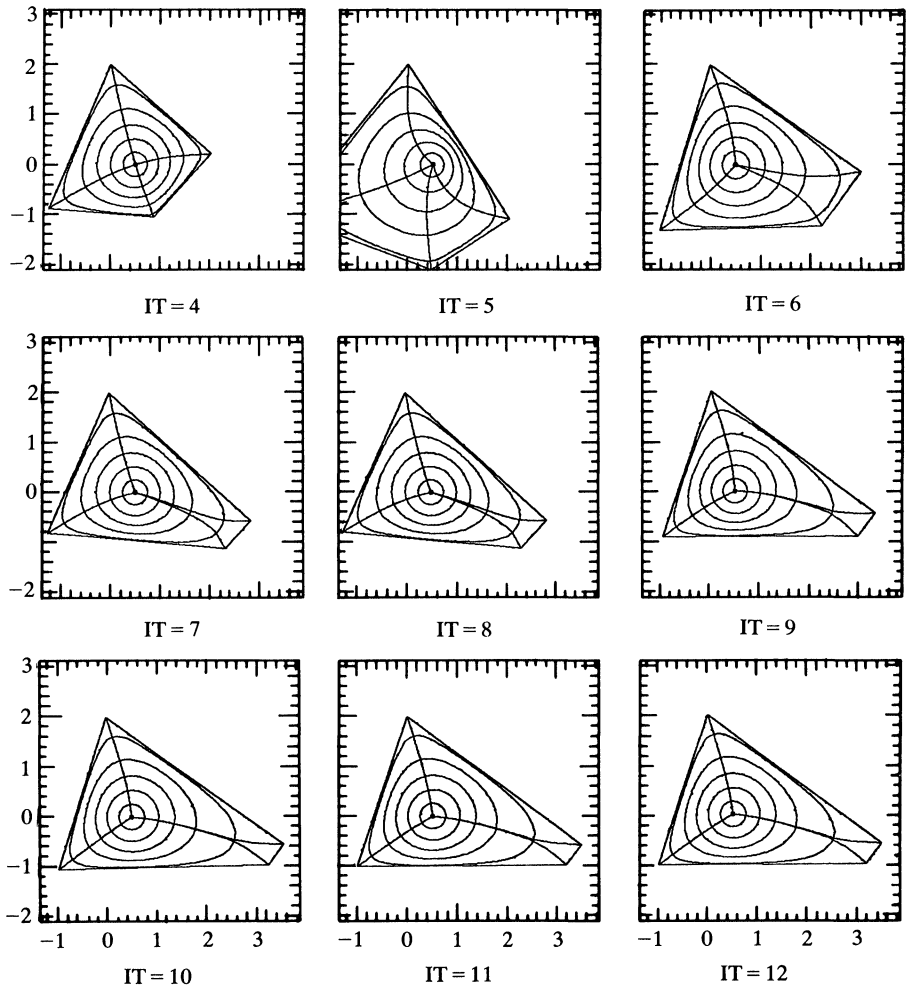


FIG. 6. *Convergence to a solution of the parameter problem. Plots show the current image polygon at each step as the accessory parameters* $\{z_k\}$ *and* C *are determined iteratively for a problem with* $N = 4$.

domain. We see in Fig. 6 that the prevertices are equally spaced around the unit circle initially (IT = 4), but move rapidly to a very uneven distribution. This behavior, which is typical, indicates why the use of a compound form of Gauss–Jacobi quadrature is so important (see § 2.3).

The sum-of-squares error in solving the nonlinear system is plotted as a function of iteration number in Fig. 7 for the same 4-vertex example. Convergence is more or less quadratic, as one would expect for Newton's method. The irregularity at iteration 19 is caused by the finite difference step size of 10^{-8} used to estimate derivatives, and would have been repeated at each alternate step thereafter if the iteration had not terminated.

5.2. Sample Schwarz–Christoffel maps. Figures 8 and 9 show plots of computed Schwarz–Christoffel maps for representative problems. The polygons of Fig. 8 are bounded and those of Fig. 9 are unbounded. Observe that contour lines bend tightly around reentrant corners, revealing the large gradients there, while avoiding the backwater regions near outward-directed corners and vertices at infinity. Like the plots

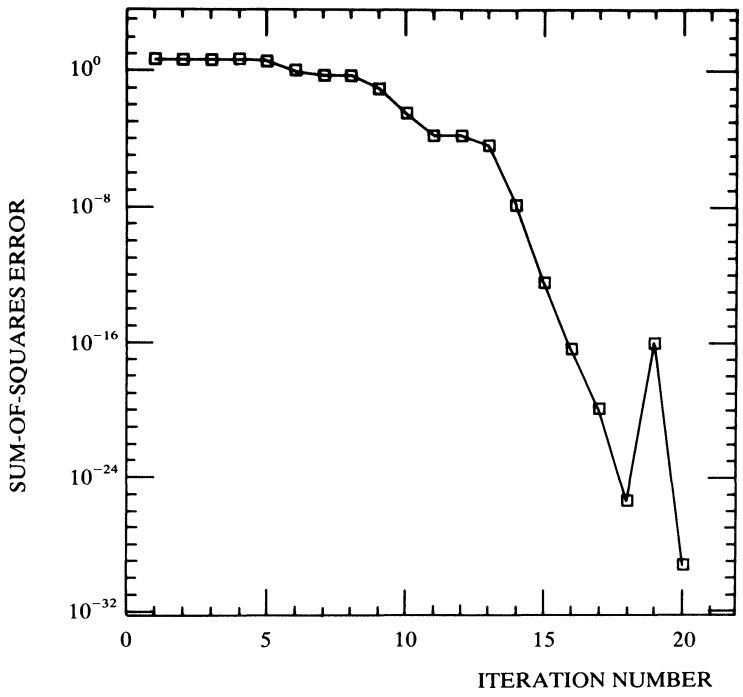


FIG. 7. *Rate of convergence. Sum-of-squares error in the nonlinear system* (2.4) *as a function of iteration number for the same problem as in Fig.* 6.

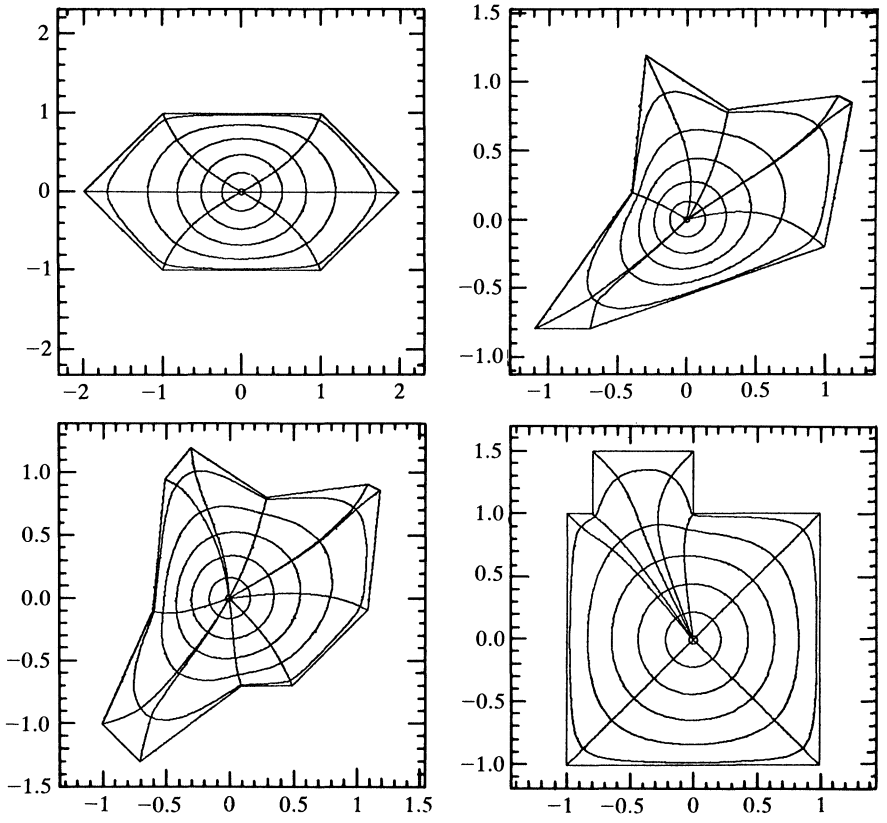


FIG. 8. *Sample Schwarz–Christoffel transformations (bounded polygons). Contours within the polygons are images of concentric circles at radii* .03, .2, .4, .6, .8, .97 *in the unit disk, and of radii from the center of the disk to the prevertices* z_k.

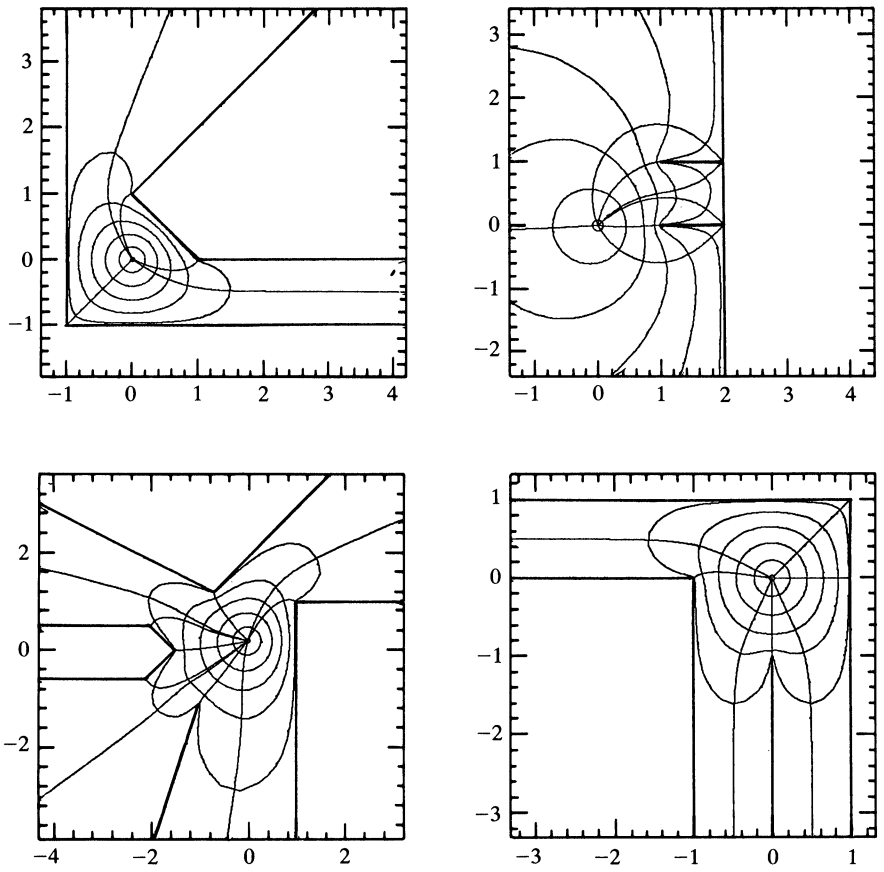


FIG. 9. *Sample Schwarz–Christoffel transformations (unbounded polygons). Contours are as in Fig. 8.*

of Fig. 6, these may be viewed as showing simultaneously the image polygon and the domain disk.

Figure 10 shows similar plots in which streamlines rather than contour lines have been plotted, so that the configuration may be thought of as portraying ideal irrotational fluid flow through a two-dimensional channel. To plot these streamlines an analytic transformation of the disk to an infinite channel with straight parallel sides was used in conjunction with the Schwarz–Christoffel transformation from the disk to the problem domain.

5.3. Discussion of applications. The usefulness of conformal mapping for applied problems stems from the fact that the Laplacian operator transforms in a simple way under a conformal map. Let $f: \mathbb{C} \to \mathbb{C}$ map a region Ω_z in the z-plane conformally onto a region Ω_w in the w-plane, and let Δ_z and Δ_w denote the Laplacian operators $\partial^2/\partial x^2 + \partial^2/\partial y^2$ and $\partial^2/\partial u^2 + \partial^2/\partial v^2$, respectively, where $z = x + iy$ and $w = u + iv$. Then we may easily show

$$\Delta_z \phi(z) = |f'(z)|^2 \Delta_w \phi(f^{-1}(w)) \tag{5.1}$$

for $\phi: \Omega_z \to \mathbb{R}$ suitably differentiable. A conformal map has $|f'(z)| > 0$ everywhere; thus from (5.1) it follows that if $\phi(z)$ is the solution to the Laplace equation $\Delta_z \phi = 0$ in Ω_z, subject to Dirichlet boundary conditions $\phi(z) = g(z)$ on the boundary Γ_z, then $\psi(w) = \phi(f^{-1}(w))$ is a solution to the Laplace equation $\Delta_w \psi = 0$ in the image region $\Omega_w = f(\Omega_z)$,

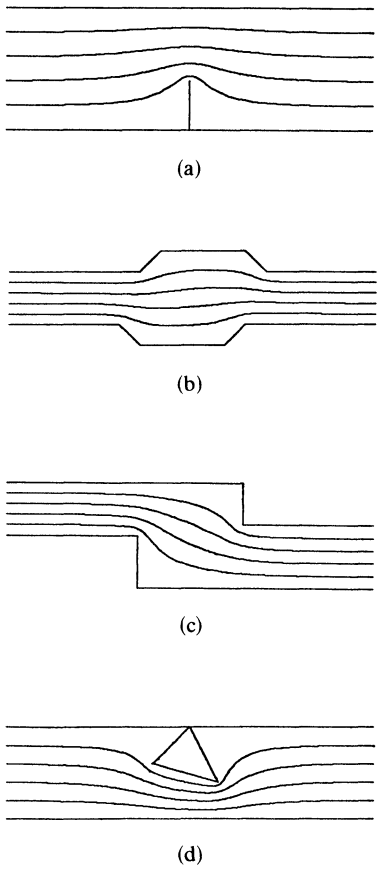


FIG. 10. *Sample Schwarz–Christoffel transformations. Contours show streamlines for ideal irrotational, incompressible fluid flow within each channel.*

subject to the image boundary conditions $\psi(w) = g(f^{-1}(w))$ on the boundary $\Gamma_w = f(\Gamma_z)$. (We have assumed that f maps Γ_z bijectively onto the boundary of Ω_w. This is not always true, but it is true if both regions are bounded by Jordan curves. See [8, Thm. 5.10e].)

More generally, from (5.1) we can see that Poisson's equation, $\Delta_z \phi(z) = \rho(z)$, transforms under a conformal transformation into a Poisson equation in the w-plane with altered right-hand side:

$$\Delta_w \psi(w) = |f'(f^{-1}(w))|^{-2} \rho(f^{-1}(w)). \tag{5.2}$$

Furthermore, more general boundary conditions than Dirichlet also transform in a simple way. For example, the Neumann condition $(\partial/\partial n_z)\phi(z) = h(z)$, where $\partial/\partial n_z$ is a normal derivative in the z-plane, transforms to $(\partial/\partial n_w)\psi(w) = |f'(f^{-1}(w))|^{-1} h(f^{-1}(w))$. We do not pursue such possibilities further here; for a systematic treatment see Chapter VI of [9].

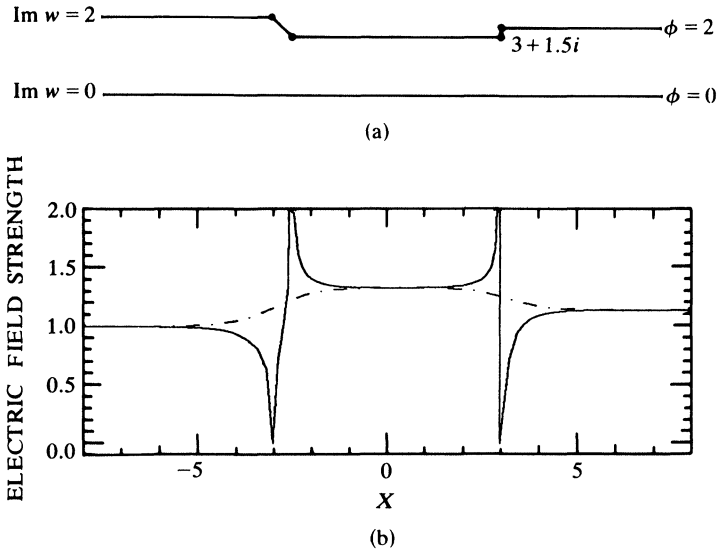
Traditionally, conformal mapping has been applied most often in two areas. One is plane electrostatics, where the electrostatic potential φ satisfies Laplace's equation. The other is irrotational, incompressible fluid flow in the plane, which may be described in terms of a velocity potential φ that also satisfies Laplace's equation. We will outline some ways in which a known conformal map might be used in such application.

Conformal maps do not solve problems, but they may reduce hard problems to easier ones. How much work must be done to solve the easier problem will vary considerably with the application.

(1) In the best of circumstances, the original problem may be reduced to a model problem whose solution is known exactly. This is the case in the fluid flow problems of Fig. 10, in which a crooked channel may be mapped to an infinite straight channel of constant width.

(2) If a problem of Laplace's equation with pure Dirichlet or Neumann boundary conditions can be mapped conformally to a disk, then Poisson's formula or Dini's formula (see [9]) provide integral representations of the solution at each interior point. Such integrals may be evaluated readily on the computer to yield high accuracy solutions. The primary disadvantage of this approach is that a new integral must be evaluated for each point at which the solution is desired.

(3) If the solution will be required at many points in the domain, then it is probably more efficient to solve Laplace's equation by a trigonometric expansion of the form $b_0+\sum_{k=1}^{m} r^k(a_k \sin k\theta + b_k \cos k\theta)$; coefficients a_k and b_k are selected so as to fit the boundary conditions closely. A disadvantage of this method is that convergence of the expansion may be slow if the boundary conditions are not smooth.



w	ϕ	$\lvert E\rvert$	$\arg E/\pi$
$3.1 + 1.4i$	1.7564	1.3082	−.3823
$3.01 + 1.49i$	1.9486	2.4403	−.2833
$3.001 + 1.499i$	1.9889	5.2137	−.2572
$3.000 + 1.500i$	2.0000	∞	−.2500

(c)

FIG. 11. *Laplace equation example: electric potential and field between two infinite sheets.*

(a) *Problem domain: region between two conducting sheets.*

(b) *Field strength along the top boundary (solid line) and bottom boundary (broken line).*

(c) *Computed potential and field strength at three points near* $3+1.5i$.

(4) Finally, if simpler methods fail, a solution in the model domain may be found by a finite-difference or finite-element technique. For problems of Poisson's equation or more complicated equations this will probably normally be necessary.

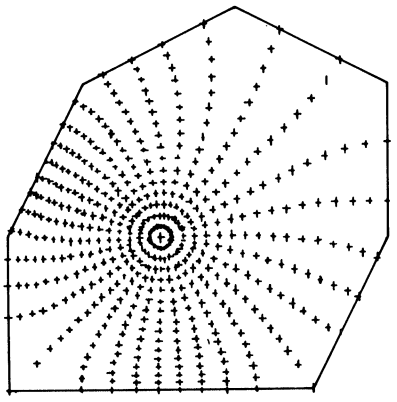
5.4. Laplace's equation. Figure 11 presents an example of type (1) as described in the last section. We are given an infinite region bounded by one straight boundary fixed at potential $\varphi = 0$ and one jagged boundary fixed at $\varphi = 2$. We may think of this as an electrostatics problem. The central question to be answered computationally will be: what are the voltage φ and the electric field $E = -\nabla\varphi$ at a given point, either within the field or on the boundary?

We proceed by mapping the given region onto the disk by a Schwarz–Christoffel transformation, then analytically onto an infinite straight channel (as in the examples of Fig. 10). In the straight channel φ and E are known trivially, and this information may be transferred to the problem domain through a knowledge of the conformal map that connects them and of its (complex) derivative. We omit the details, which are straightforward.

Figure 11(b) shows $|E|$ as a function of x on the upper and lower boundaries of the region. To see more of the behavior of the solution field near a reentrant corner, we also compute the field at three points near $3+1.5i$. These results are given in Fig. 11(c).

5.5. Poisson's equation. Consider the 7-sided region shown in Fig. 12(a). We wish to solve Poisson's equation

$$\Delta\phi(x, y) = \tfrac{1}{5}\sin 2x(1-2(y+1)^2)$$



(a)

Grid ($r\times\theta$)	Transformation and setup time	Fast Poisson solver time	Max. error	RMS error
4×8	1.3 secs.	<.01 secs.	0.132	0.0309
8×16	2 secs.	.01 secs.	0.055	0.0085
16×32	5 secs.	.03 secs.	0.031	0.0037
32×64	16 secs.	.15 secs.	0.026	0.0012

(b)

FIG. 12. *Poisson equation example. Problem is transplanted conformally to the unit disk and solved by finite differences.*

(a) *7-sided problem domain, including image of* 16×32 *finite-difference grid in the unit disk.*

(b) *Computed results for four different grids. Time estimates are for an IBM* 370/168.

on this region, subject to Dirichlet conditions

$$\phi(x, y) = \rho(x, y) = \tfrac{1}{10} \sin 2x(y+1)^2$$

on the boundary. We proceed by mapping the domain to the disk and solving a transformed problem in the disk in polar coordinates by means of a second-order fast finite difference solver (PWSPLR, by P. Swarztrauber and R. Sweet). $\rho(x, y)$ is the correct solution in the interior as well as on the boundary, so we can determine the accuracy of the numerical solution.

This is not as satisfactory a procedure as was available for Laplace equation problems. According to (5.2), the model problem here is Poisson's equation in the disk with an altered right-hand side containing the factor $|f'(z)|^2$, where f is the composite map from the disk to the 7-gon. Two difficulties arise. The first is that to set up the transformed equation in the disk, $\rho(w_{ij})$ must be computed for every $w_{ij} = w(z_{ij})$ which is an image of a grid point in the disk. This is time consuming, one hundred times more so in this experiment than the fast solution of Poisson's equation once it is set up. Second, $|f'(z)|^2$ is singular (unbounded, in this example) at each prevertex z_k, and this appears to interfere with the second-order accuracy which we would like to observe. The table in Fig. 12(b) attests to both of these problems.

5.6. Eigenfrequencies of the Laplace operator. Petter Bjørstad (Computer Science Dept., Stanford University) has recently combined the present Schwarz–Christoffel computation with a fast finite-difference scheme to successfully compute eigenvalues and eigenvectors of the Laplacian operator on polygonal regions. These results may be interpreted as giving the normal modes and frequencies of a thin membrane in two dimensions, or of a three-dimensional waveguide with constant cross-section. This work will be reported elsewhere.

6. Conclusions. A program has been described which computes accurate Schwarz–Christoffel transformations from the unit disk to the interior of a simply connected polygon in the complex plane, which may be unbounded. Key features of the computation have been:

(1) choice of the unit disk rather than the upper half-plane as the model domain, for better numerical scaling (§ 2.1);
(2) use of complex contour integrals interior to the model domain rather than along the boundary, making possible the treatment of unbounded polygons (§ 2.1);
(3) use of compound Gauss–Jacobi quadrature in complex arithmetic to evaluate the Schwarz–Christoffel integral accurately (§§ 2.3, 3.1);
(4) formulation of the parameter problem as a constrained nonlinear system in $N-1$ variables (§ 2.1);
(5) elimination of constraints in the nonlinear system by a simple change of variables (§ 2.2);
(6) solution of the system by a packaged nonlinear systems solver; no initial estimate required in practice (§ 2.4);
(7) computation of a reliable estimate of the accuracy of further computations, once the parameter problem has been solved (§ 4.1);
(8) accurate evaluation of the inverse mapping in two steps by means of a packaged o.d.e. solver and Newton's method (§ 3.2).

Previous efforts at computing Schwarz–Christoffel transformations numerically include [1], [6], [7], [10], and [13]. The present work differs from these in that it deals directly with complex arithmetic throughout, taking the unit disk rather than the upper

half-plane as the model domain and evaluating complex contour integrals. This makes possible the computation of transformations involving general unbounded polygons. (Cherednichenko and Zhelankina [1] also treat unbounded polygons, by a different method.) Two other important differences are the use of compound Gauss–Jacobi quadrature, and the application of a change of variables to eliminate constraints in the nonlinear system ((5), above). We believe that our program computes Schwarz–Christoffel transformations faster, more accurately, and for a wider range of problems tha previous attempts.

A variety of directions for further work suggest themselves. Here are some of them.

(1) More attention should be paid to the problems of evaluating the forward and inverse S–C maps once the parameter problem has been solved. The two-step method for the inverse map described in § 3.2 is reliable, but it uses too much machinery. Recently Petter Bjørstad and Eric Grosse of Stanford University have replaced (3.1) with a power series expansion for problems in which all the nodes of a finite-difference grid must be mapped from one domain to the other, thereby speeding up the evaluations of $w(z)$ and $z(w)$ by an order of magnitude. This kind of addition is very important for applications.

(2) The program could easily be extended to construct maps onto the exterior of a polygon—that is, the interior of a polygon whose interior includes the point at infinity. This extension would be necessary, say, for applications to airfoil problems.

(3) It should not be too great a step to raise the present program to the level of "software" by packaging it flexibly, portably, and robustly enough that naive users could apply it easily to physical problems. Conformal mapping is currently far behind many other areas of numerical mathematics in the development and distribution of software.

(4) The program might be extended to handle the rounding of corners in Schwarz–Christoffel transformations (see [8]). What about mapping doubly or multiply connected polygonal regions, perhaps by means of an iterative technique which computes an S–C transformation at each step?

(5) More generally, the Schwarz–Christoffel formula should be viewed in context as a particularly simple method in conformal mapping which is applicable only to a limited set of geometries. Direct comparisons with programs that can treat curved boundaries, especially those based on integral equations, would be informative. In some applications the S–C transformation might profitably be used as part of a larger program. In fact, the S–C formula (1.2) itself has a natural generalization to the case of curved boundaries, which may be obtained formally by allowing an infinite number of vertices with infinitesimal external angles. R. T. Davis [3] has implemented this formula numerically with very promising results.

Most important, further work is needed in the direction of applications to Laplace's equation, Poisson's equation, and related problems. Irregular or unbounded domains are generally troublesome to deal with by standard techniques, particularly when singularities in the form of reentrant corners are present. Schwarz–Christoffel transformations offer a means of getting around such difficulties in a natural way. More experience is needed here.

Note. This work is described in more detail in [12], and a program listing is given there. An experimental copy of the package with documentation and sample driver programs may be obtained from the author.

Acknowledgments. This work was suggested and guided by Peter Henrici, without whom it would not have been possible. Computations were performed at the Stanford Linear Accelerator Center of the U.S..Department of Energy. The author has benefited from discussions with Petter Bjørstad, William M. Coughran, Jr., Gene Golub, Eric Grosse, Randy LeVeque, and Cleve Moler.

REFERENCES

[1] L. A. CHEREDNICHENKO AND I. K. ZHELANKINA, *Determination of the constants that occur in the Christoffel–Schwarz integral.* Izv. Vyssh. Ucebn. Zaved. Elektromehanika, 10 (1975), pp. 1037–1040. (In Russian.)

[2] P. J. DAVIS AND P. RABINOWITZ, *Methods of Numerical Integration*, Academic Press, New York, 1975.

[3] R. T. DAVIS, *Numerical methods for coordinate generation based on Schwarz–Christoffel transformations*, 4th AIAA Computational Fluid Dynamics Conference Proceedings, Williamsburg, VA, 1979.

[4] D. GAIER, *Konstruktive Methoden der konformen Abbildung*, Springer-Verlag, Berlin, 1964.

[5] G. H. GOLUB AND J. H. WELSCH, *Calculation of Gaussian Quadrature Rules*, Math. Comp., 23 (1969), pp. 221–230.

[6] T. R. HOPKINS AND D. E. ROBERTS, *Kufarev's Method for the Numerical Determination of the Schwartz–Christoffel Parameters*, University of Kent, Kent, England, 1978.

[7] D. HOWE, *The application of numerical methods to the conformal transformation of polygonal boundaries*, J. Inst. Math. Appl., 12 (1973), pp. 125–136.

[8] P. HENRICI, *Applied and Computational Complex Analysis I*, Wiley, New York, 1974.

[9] L. V. KANTOROVICH AND V. I. KRYLOV, *Approximate Methods of Higher Analysis*, P. Noordhoff, Groningen, the Netherlands, 1958.

[10] E.-S. MEYER, *Praktische Verfahren zur konformen Abbildung von Geradenpolygonen.* Dissertation Universität Hannover, 1979.

[11] M. J. D. POWELL, *A Fortran subroutine for solving systems of non-linear algebraic equations*, Tech. Rep. AERE-R. 5947, Harwell, England, 1968.

[12] L. N. TREFETHEN, *Numerical computation of the Schwarz–Christoffel transformation*, Computer Science Dept. Tech. Rep. STAN-CS-79-710, Stanford Univ., Stanford, CA, 1979.

[13] V. V. VECHESLAVOV AND V. I. KOKOULIN, *Determination of the parameters of the conformal mapping of simply connected polygonal regions*, U.S.S.R. Computational Math. and Math. Phys., 13 (1974), pp. 57–65.