SIAM J. SCI. STAT. COMPUT.
Vol. 1, No. 3, September 1980

© 1980 Society for Industrial and Applied Mathematics
0196-5204/80/0103-0007 $01.00/0

A NUMERICAL METHOD FOR CONFORMAL MAPPINGS*

BENGT FORNBERG†

Abstract. A numerical technique is presented for calculating the Taylor coefficients of the analytic function which maps the unit circle onto a region bounded by any smooth simply connected curve. The method involves a quadratically convergent outer iteration and a super-linearly convergent inner iteration. If N complex points are distributed equidistantly around the periphery of the unit circle, their images on the edge of the mapped region, together with approximations for the $N/2$ first Taylor coefficients, are obtained in $O(N \log N)$ operations. A calculation of time-dependent waves on deep water is discussed as an example of the potential applications of the method.

Key words. conformal mapping, Fast Fourier transform, conjugate gradients, water waves

1. Introduction. A large number of numerical methods have been proposed for calculating approximations to the unique mapping which is described in the Riemann mapping theorem. We will consider the problem of finding the leading Taylor coefficients in a mapping from the unit circle to a given simply connected domain bounded by a smooth curve J.

A book by Gaier [3] gives detailed descriptions of the methods known in 1964. The introduction of the Fast Fourier Transform algorithm (FFT) shortly afterwards (Cooley and Tukey [2], 1965) made possible dramatic increases in the efficiency of some of these methods (Henrici [7]). The most important of the currently used methods seem to be different approximations of Theodorsen's integral equation (Henrici [7], Gutknecht [4], [5]), Symm's method (Symm [10], Henrici [7], Hayes et al. [6]), a method based on numerical solution of the Cauchy-Riemann equations in conjunction with optimization techniques (Chakravarthy and Anderson [1]) and a method based on some new integral equations (Menikoff and Zemack [9]). Theodorsen's method is limited to regions which have single-valued representations in polar coordinates. For almost all such regions, good performance also requires estimates for certain relaxation parameters. The position of the N points on a fixed curve J, which correspond to the N roots of unity in the mapping from the unit circle, can be found in $O(N \log N)$ operations. However, the proportionality constant depends strongly on the shape of J. Symm's method has a similar operation count for simple regions, but in addition, allows general regions with an increase in operation count to $O(N^3)$ (Hayes et al. [6]). The method of Chakravarthy and Anderson [1] requires $O(N^3)$ operations if a Newton optimization technique is used, but may go somewhat faster with an alternative conjugate gradient procedure. The method by Menikoff and Zemach also costs $O(N^3)$ operations, but it allows an arbitrary distribution of the computational points on the boundary.

Most of the methods address the problem of finding a mapping to or from a unit circle. They establish first the mapping of the boundaries. From this follows then the complete mapping function. The method we will present produces approximations to the Taylor coefficients in the mapping from the unit circle onto the given region at the same time as it finds the boundary correspondence. For this reason, we will describe it as a method to map the unit circle onto a given region rather than a method for the

*Received by the editors January 24, 1980. This research was supported by Control Data Corporation and by the U.S. Department of Energy Office of Basic Energy Sciences.

†Department of Applied Mathematics, California Institute of Technology, Pasadena, California 91125.

inverse mapping. The method has an operation count of $O(N\log N)$ for the general case, i.e., without any restrictions on the region to be "near-circular" or "starshaped." The convergence rate appears to be only weakly affected by the complexity of the region and no parameters are required to optimize the performance. Before we describe the idea of the method, we will briefly illustrate some possible ill-conditioning of the mapping problem and mention the application in fluid mechanics for which this method was originally developed.

Fig. 1 illustrates a typical case of mapping from the unit circle to a simple region (described in the last section as test case 1 with $\alpha = .5$). It transpires in this case that more than 8000 Taylor coefficients of the mapping function are needed to obtain the mapping with a 10^{-8} accuracy. This in turn requires N, the number of points on the periphery, to exceed 16,000. The density of the mapped points (uniform on the unit circle) varies by as much as a factor of 400 along different parts of the edge. Very small changes in the shape of the region will quite dramatically change the position of some boundary points. It is clear that conformal mapping is an ill-conditioned problem. One might try to avoid part of this difficulty by looking for a more economical functional representation of the mapping function than a Taylor series. Simply moving the origin would help the economy in this case, but would not help in many others. For example, in cases of fixed shapes like airfoils, initial explicit transforms can be used to first remove a corner or a cusp. A preliminary sequence of transformations, for example with square root branch points just outside indentations in J, may be useful to make the region closer to a circle. Even with the implementation of such preliminary steps we must at some stage find the remaining mapping between a near-circular region and a perfect circle.

Applications of conformal mappings include generation of computational grids and simplifications of geometries for analytical work (for, example, to find electrical fields or potential flows around bodies). Which method is most suitable depends on the application, in particular on the type of geometry (with or without corners, near-circular shape or not, etc.) and the importance of computational efficiency (mapping performed only once or performed repeatedly). The mapping method described in this paper was developed particularly for the calculation of time-dependent waves on inviscid and irrotational deep water. We consider a periodic section of deep water, as shown to the left in a complex $z = x + iy$ -plane in Fig. 2, and introduce a velocity potential $\phi(x, y, t)$ such that ϕ_x and ϕ_y are the x- and y-velocities

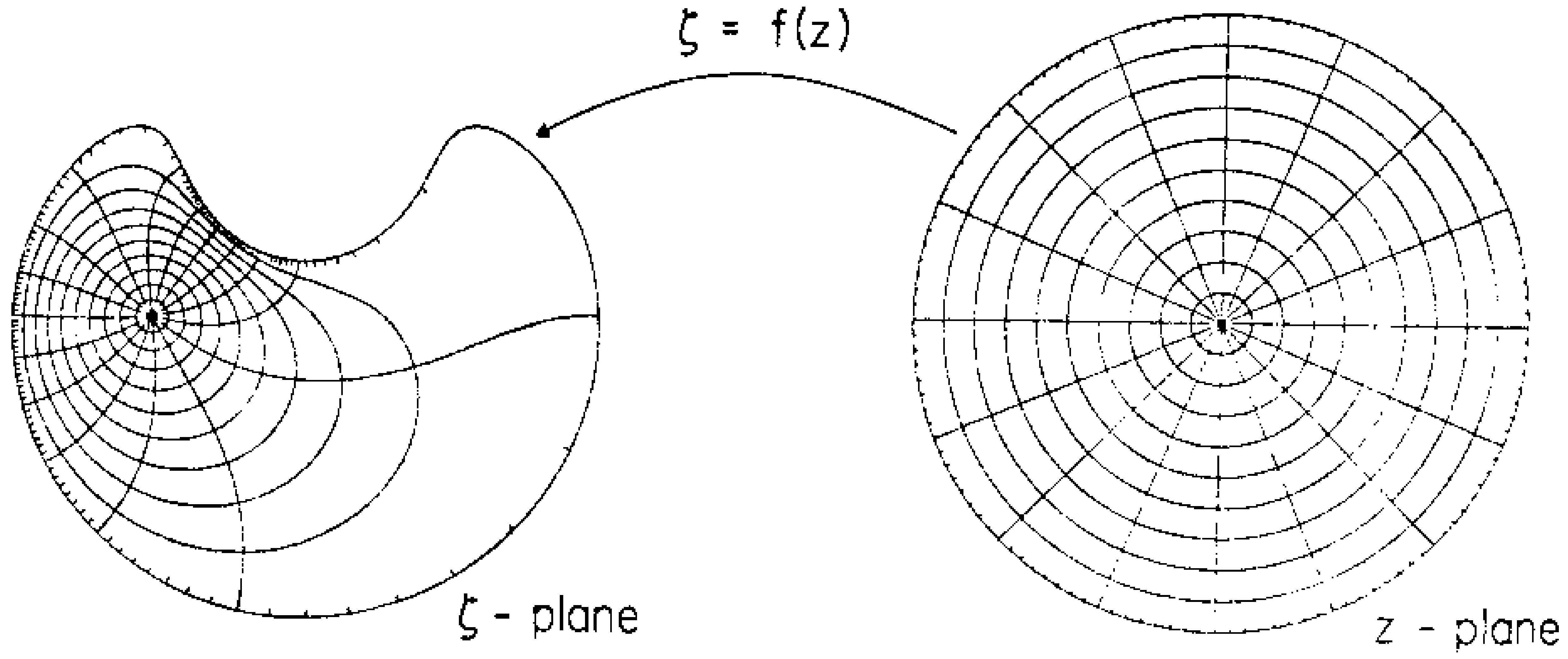

FIG 1. *Example of conformal mapping.*

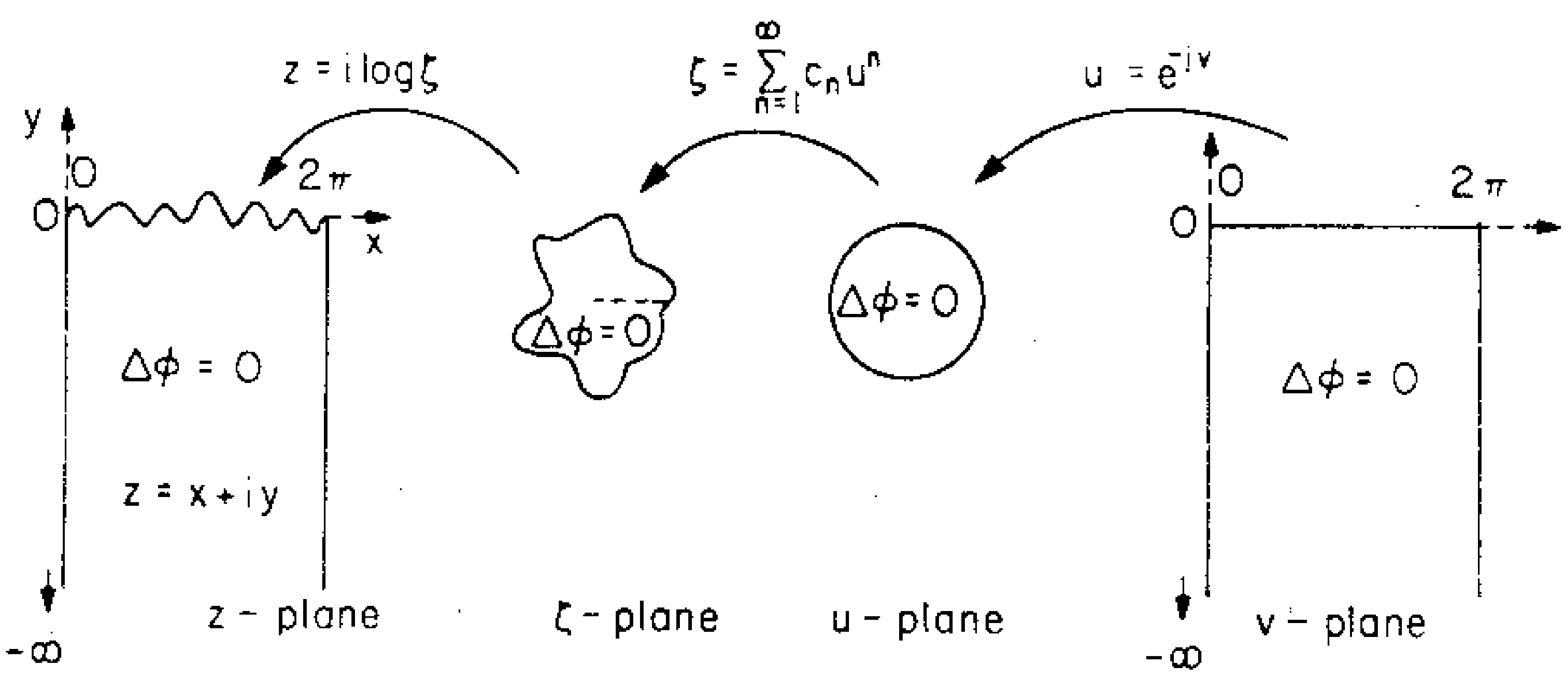

FIG 2. *The steps in a conformal mapping of a wavy surface on deep water to a fixed, flat surface.*

of the fluid elements. The governing equation is

$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\phi=0, \tag{1}$$

inside the fluid, with two conditions on the free surface. One is

$$\frac{\partial\phi}{\partial t}=-gy-\tfrac{1}{2}\left(\phi_x^2+\phi_y^2\right), \tag{2}$$

(where g is a gravity constant) and the other one expresses the fact that fluid elements on the surface remain on the surface. Fig. 2 illustrates how, at any step in a numerical time marching, the wavy surface can be mapped to a flat one, leaving the form of (1) unchanged. The derivatives ϕ_x and ϕ_y, required for a time step, can now be obtained easily. In one test calculation, a second order modified Euler scheme was used to advance a uniformly traveling periodic wavetrain five periods in space. The initial condition was a Stokes' wave with height over wavelength .09083 (cf. maximal wave .14107) and a theorectical speed of 1.04155. Fig. 3 shows the solution at time $t=30.1628$ superposed on the initial wave. The very small discrepancy between the curves at the top of the wave is caused by graphical straight line interpolation between the 32 poiints used for the spatial discretization.

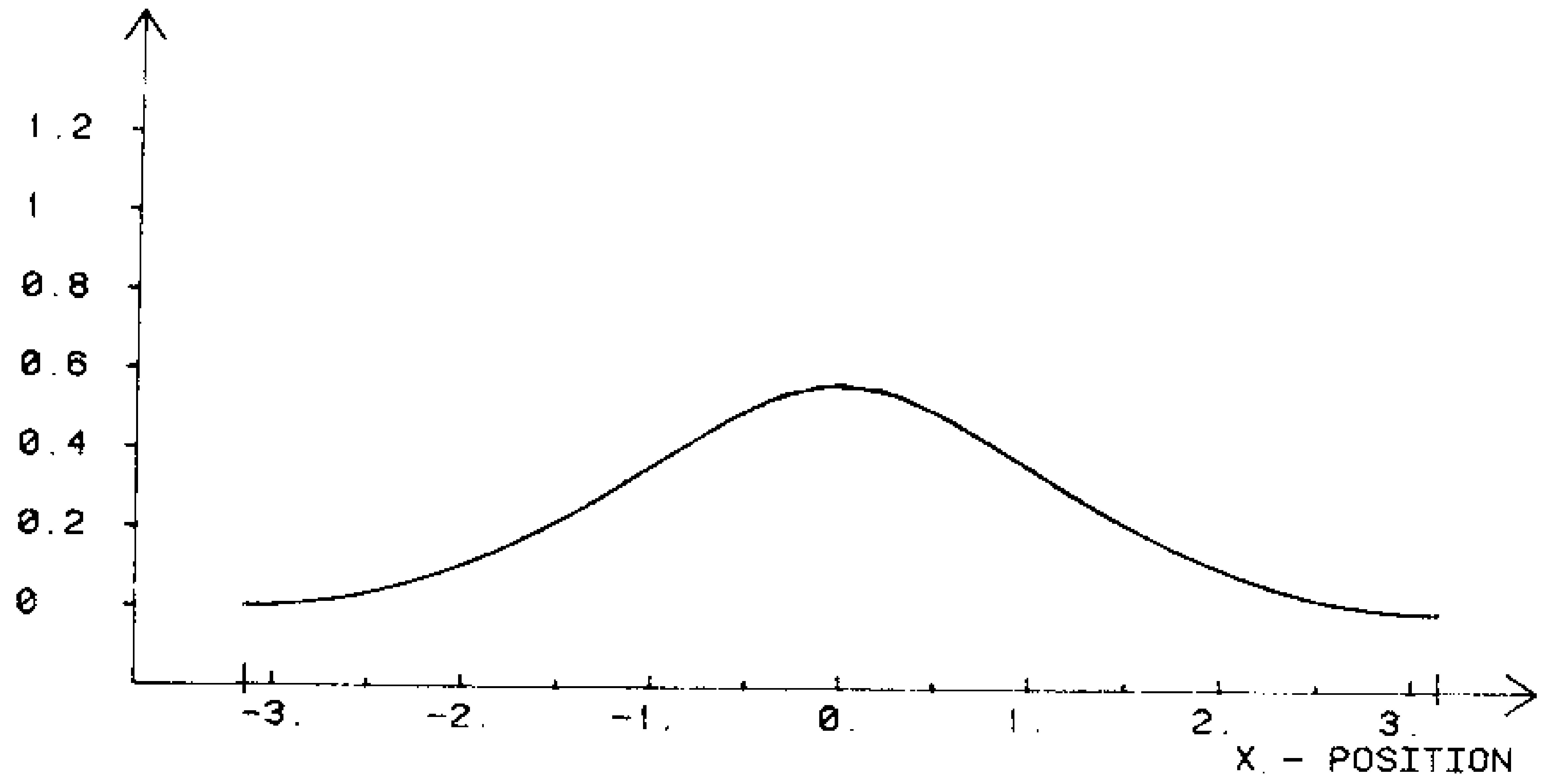

FIG 3. *The initial shape of a Stoke's wave compared with the calculated shape and position after five periods. (Spatial resolution 32 points.)*

The idea of the mapping method can be described as follows: we introduce N complex points ζ_i ordered monotonically along the boundary curve J. The problem is to find a strategy for moving these points along J so that, through the unique mapping (which is unknown to us), they will come to correspond to the N roots of unity z_i on the unit circle. For each guess of the positions of the points ζ_i, an analytic function

$$(3) \qquad \zeta(z)=\sum_{\nu=-(N/2)+1}^{N/2} d_\nu z^\nu$$

is introduced. The coefficients d_ν are obtained from a complex discrete Fourier transform applied to the numbers ζ_i. The function $\zeta(z)$ becomes one particular case of an analytic function which maps the points z_i into the points ζ_i on J. However, this function $\zeta(z)$ is in general singular for $z=0$. This is an unacceptable property of a mapping function which is required to satisfy $\zeta(0)=0$. We will describe a quadratically convergent strategy to move all the points ζ_i along the curve J in such a way that the function $\zeta(z)$ loses its singularities inside the unit circle and will satisfy $\zeta(0)=0$. In other words, we will make $d_\nu=0$ for $\nu=0,-1,-2,\ldots,-N/2+1$. The coefficients d_ν for $\nu=1,2,3,\ldots,N/2$ will then approximate the Taylor coefficients of the mapping function. Each step in this process gives rise to a linear problem which is solved by a (super-linearly converging) conjugate gradient iteration.

This mapping method has been coded in vectorized Fortran on a CDC STAR-100 computer (located at the Control Data Corporation Service Center in Arden Hills, Minnesota). We wish to express our gratitude to Control Data Corporation for making their STAR-100 computer system available for this work.

2. Description of the outer iteration. The boundary curve J is assumed to be smooth, simply connected and enclosing the origin in a complex plane. Since J is smooth, the true mapping function $\xi(z)$ can be represented by a convergent Taylor series as

$$(4) \qquad \xi(z)=\sum_{\nu=1}^{\infty} c_\nu z^\nu, \qquad |z|\leqq 1.$$

(We denote by $\zeta(z)$ the approximate mapping function we will determine). One more condition than $\xi(0)=0$ (and $\zeta(0)=0$) is needed to uniquely determine the mapping. The condition $\partial\xi/\partial z>0$ is often used. We will instead require that $\zeta(1)$ lies at a specified point on J. For values of z on the unit circle, i.e., $z(\theta)=e^{2\pi i\theta}$, $0\leqq\theta\leqq 1$, (4) becomes

$$(5) \qquad \xi(z(\theta))=\sum_{\nu=1}^{\infty} c_\nu e^{2\pi i\nu\theta}.$$

This is called the "boundary correspondence function," a uniquely determined periodic function of θ. We consider now (5) at the θ-values $\theta_k=k/N$, $k=0,1,\ldots,N-1$, and suppose that N is even (N a power of 2 is most efficient and will be assumed in our operation count). We get

$$(6) \qquad \xi_k=\xi(z(\theta_k))=\sum_{\nu=-(N/2)+1}^{N/2} g_\nu e^{2\pi i\nu k/N}=\sum_{\nu=-(N/2)+1}^{N/2} g_\nu w^{\nu k},$$

where $w=e^{2\pi i/N}$ and

$$(7) \qquad g_\nu=\sum_{j=0}^{\infty} c_{\nu+jN}.$$

defining $c_\nu = 0$ for $\nu \leqq 0$. The error in accepting g_ν as an approximation for c_ν is

$$g_\nu - c_\nu = \sum_{j=1}^{\infty} c_{\nu+jN}. \tag{8}$$

N can now be chosen so large that these errors for $\nu = 1, 2, \ldots, N/2$ are within our tolerance.

The discrete Fourier transform in (6) can be inverted

$$g_\nu = \frac{1}{N} \sum_{k=0}^{N-1} \xi_k w^{-k\nu}, \qquad \nu = -\frac{N}{2} + 1, \ldots, \frac{N}{2}. \tag{9}$$

Hence, given the positions for the N points ξ_k, we obtain all the coefficients g_ν by applying one FFT. By choosing N sufficiently large, the values of g_ν, $\nu = -N/2 + 1, \ldots, 0$, become arbitrarily small, and those of g_ν, $\nu = 1, \ldots, N/2$ arbitrarily close to the corresponding c_ν. This leads us to consider the following approximate analogue of (9):

$$d_\nu = \frac{1}{N} \sum_{k=0}^{N-1} \zeta_k w^{-k\nu}, \qquad \nu = -\frac{N}{2} + 1, \ldots, \frac{N}{2}. \tag{10}$$

Here, the points ζ_k lie monotonically along J and represent guesses for the numbers ξ_k. We wish to move these points ζ_k on J in such a way that d_ν becomes equal to zero for $\nu = -N/2 + 1, \ldots, 0$. With N free real parameters, we wish to make $N/2$ complex numbers zero. This count of equations and unknowns appears correct, but we have not yet prescribed a position to one of the points. It will transpire that the N equations we obtain after linearization will form a system with rank only $N-1$ (to within truncation errors).

We move the points ζ_k in a two-step process. Given the tangential directions e_k (with $|e_k| = 1$) at the points ζ_k on J, we can try to move these points in the tangential directions by distances t_k in such a way that $d_0, d_{-1}, \ldots, d_{-N/2+1}$ become zero:

$$0 = \frac{1}{N} \sum_{k=0}^{N-1} (\zeta_k + t_k e_k) w^{-k\nu}, \qquad \nu = -\frac{N}{2} + 1, \ldots, 0. \tag{11}$$

Afterwards, the points $\zeta_k + t_k e_k$ are moved back to the curve J. Since J is smooth, the distance from the curve is $O(t_i^2)$. From this follows the quadratic convergence of this outer iteration.

By subtracting the νth equation in (11) from the νth equation in (10) we get $N/2$ complex linear equations for the N real unknowns t_k:

$$d_\nu = -\frac{1}{N} \sum_{k=0}^{N-1} t_k e_k w^{-k\nu}, \qquad \nu = -\frac{N}{2} + 1, \ldots, 0, \tag{12a}$$

or, using matrix notation:

(12b)

$$\begin{bmatrix} d_0 \\ d_{-1} \\ d_{-2} \\ \vdots \\ d_{-N/2+1} \end{bmatrix} = -\frac{1}{N} \left[\begin{array}{cccccc:cc} 1 & 1 & 1 & 1 & \cdots & & \cdots & 1 \\ 1 & w & w^2 & w^3 & & & & w^{N-1} \\ 1 & w^2 & w^4 & w^6 & & & & w^{2N-2} \\ \vdots & & & & & & & \vdots \\ 1 & w^{N/2-1} & w^{N-2} & w^{3N/2-3} & \cdots & & \cdots & w^{(N/2-1)(N-1)} \end{array} \right] \begin{bmatrix} e_0 t_0 \\ e_1 t_1 \\ e_2 t_2 \\ \\ \\ \vdots \\ \\ e_{N-1} t_{N-1} \end{bmatrix}.$$

Two questions must now be investigated.

1. Can the special structure of the coefficient matrix in the linear system (12) be exploited to give a very fast method of solution?

2. The position of one point, for example ζ_0, should be arbitrary. How is this freedom present in (12)? These questions are discussed in the next three sections. A practical implementation of the method and its performance on some test cases is described in the final section.

3. Reformulation of the linear system of equations. Collecting the odd- and even-numbered columns of the matrix in (12b) gives

$$-\begin{bmatrix} d_0 \\ d_{-1} \\ d_{-2} \\ \vdots \\ d_{-N/2+1} \end{bmatrix} = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w^2 & w^4 & & w^{N-2} \\ 1 & w^4 & w^8 & & w^{2N-4} \\ \vdots & & & & \vdots \\ 1 & w^{N-2} & w^{2N-4} & \cdots & w^{2(N/2-1)^2} \end{bmatrix} \times \begin{bmatrix} e_0 & & & \\ & e_2 & & \\ & & e_4 & \\ & & & \ddots \\ & & & & e_{N-2} \end{bmatrix} \times \begin{bmatrix} t_0 \\ t_2 \\ t_4 \\ \vdots \\ t_{N-2} \end{bmatrix}$$

$$\tag{13} +\frac{1}{N}\begin{bmatrix} w^0 & & & & \\ & w^1 & & & \\ & & w^2 & & \\ & & & \ddots & \\ & & & & w^{N/2-1} \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w^2 & w^4 & & w^{N-2} \\ 1 & w^4 & w^8 & & w^{2N-4} \\ \vdots & & & & \vdots \\ 1 & w^{N-2} & w^{2N-4} & \cdots & w^{2(N/2-1)^2} \end{bmatrix}$$

$$\times \begin{bmatrix} e_1 & & & & \\ & e_3 & & & \\ & & e_5 & & \\ & & & \ddots & \\ & & & & e_{N-1} \end{bmatrix} \times \begin{bmatrix} t_1 \\ t_3 \\ t_5 \\ \vdots \\ t_{N-1} \end{bmatrix},$$

or

$$\tag{14} -\mathbf{d} = \frac{1}{N} FE_0\mathbf{t}_0 + \frac{1}{N} WFE_1\mathbf{t}_1.$$

Here, F is the discrete Fourier transform matrix of order $N/2$. The matrices $1/\sqrt{N/2}\ F, E_0, E_1$ and W are all unitary. Multiplication of any vector by any of these matrices or their inverses will require at most $O(N \log N)$ operations.

Let us consider the following iteration: Given any real vector $\mathbf{t}_0{}^{(0)}$, solve (14) for the uniquely determined complex vector $\mathbf{t}_1{}^{(0)}$, remove the imaginary parts from $\mathbf{t}_1{}^{(0)}$ and then solve (14) for $\mathbf{t}_0{}^{(1)}$ where the imaginary parts that are obtained are again removed.

The real vector $\mathbf{t}_0{}^{(1)}$ that results from one step of this iteration will depend linearly on the initial vector $\mathbf{t}_0{}^{(0)}$. There must therefore be a relation

$$\tag{15} \mathbf{t}_0{}^{(1)} = A\mathbf{t}_0{}^{(0)} + \mathbf{b}.$$

Straightforward algebra gives now

$$(16)\qquad A=R^TR,\quad \text{where } R=\operatorname{Re} C,\quad C=\frac{2}{N}E_0^HF^HWFE_1.$$

A is symmetric and positive semidefinite and C is a unitary complex matrix. Since $\|A\|_2=\|R\|_2^2\leqq\|C\|_2^2=1$, the eigenvalues λ_i of A satisfy

$$(17)\qquad 0\leqq\lambda_i\leqq 1,\qquad i=0,1,\dots,\frac{N}{2}-1.$$

If there is a real solution $\mathbf{t}_0,\mathbf{t}_1$ of (14), $\mathbf{t}_0^{(0)}=\mathbf{t}_0$ must imply $\mathbf{t}_0^{(1)}=\mathbf{t}_0$ in (15). Hence, $\mathbf{t}_0$ must then satisfy

$$(18)\qquad G\mathbf{t}_0=\mathbf{b},$$

where $G=I-A$ is again positive semidefinite. Once $\mathbf{t}_0$ has been computed, $\mathbf{t}_1$ follows from (14). The calculation of $\mathbf{t}_0{}^{(1)}$ from $\mathbf{t}_0{}^{(0)}$, in particular the calculation of $\mathbf{b}$ by starting with $\mathbf{t}_0{}^{(0)}=0$ in (15), and the multiplication of any real vector by G is performed in $O(N\log N)$ operations if N is a power of 2.

We will now further investigate the eigenvalues of G and describe how the conjugate gradient method can be applied very efficiently to solve a modification of the system (18).

4. Eigenvalues of the *G*-matrix. The outer iteration was designed to force to zero not only the coefficients $d_{-1},d_{-2},\dots,d_{-N/2+1}$ but also d_0, thereby ensuring $\zeta(0)=0$. The mapping is still arbitrary with respect to a rotation. In particular, we should be able to require $t_0=0$, where t_0 is the first component of the vector $\mathbf{t}_0$. That means that the point ζ_0 is not moved during the mapping process. Imposing $t_0=0$ is only consistent with (18) if G is singular and if $\mathbf{b}$ lies in the subspace spanned by the columns of G except the first one. This was true (to within truncation accuracy) in all cases we tested. The following examples illustrate this result, and show typical distributions for the remaining eigenvalues.

Example 1. Mapping of the unit circle onto itself. The points ζ_j, $j=0,1,\dots,N-1$, become equidistantly spaced in this trivial mapping. Omitting a complex factor of unit magnitude (corresponding to a rotation), the tangential direction at the point ζ_j is

$$(19)\qquad e_j=e^{2\pi ij/N}.$$

The elements of C become in general

$$(20)\qquad c_{m,n}=\frac{1}{N}\{1+i\cot p\pi\}\frac{e_{2n-1}}{e_{2m-2}},$$

where $p=[2(n-m)+1]/N$. In this case, this simplifies to

$$(21)\qquad c_{m,n}=\frac{1}{N}\{-1+i\cot p\pi\}.$$

The elements of the matrix R are

$$(22)\qquad r_{m,n}=-\frac{1}{N},$$

independently of m and n. The matrix A has one eigenvalue equal to one and all the others equal to zero. Therefore, the matrix G has one eigenvalue equal to zero and the others equal to one.

The bottom line in (13) gives in this case

$$-d_{-(N/2)+1}=(t_0+t_2+\cdots+t_{N-2})-(t_1+t_3+\cdots+t_{N-1}). \tag{23}$$

Since all the t_i are real, the requirement for a solution is that $\operatorname{Im} d_{-(N/2)+1}=0$. This is satisfied since, in the mapping, all coefficients except the first one are absent. It is possible to add the same constant to all t_i in (13) and (18). This corresponds to a rotation of the mapping. Fixing $t_0=0$ removes this ambiguity.

Example 2. We consider regions bounded by curves given in the complex plane $\zeta=x+iy$ by

$$f_1(x,y,\alpha)\equiv((x-.5)^2+(y-\alpha)^2)(1-(x-.5)^2-y^2)-.1=0. \tag{24}$$

For $\alpha=\infty$, this defines a circle with center at $x=.5, y=0$. Fig. 4 shows the curves for some different values of α down to $\alpha\approx.2746687749$, at which point the region ceases to be simply connected. Fig. 5 illustrates, for different α, the distribution of the eigenvalues of G (and of the matrix $\hat{G}$ which we will introduce below).

In every example we have studied, the eigenvalues of the G show the same pattern. One eigenvalue is zero to truncation error accuracy and all others lie in a heavy cluster around $\lambda=1$. A few double eigenvalues gradually move toward smaller values of λ as the complexity of the curve increases. Increasing the number of points N for a fixed α made (to within truncation accuracy) no difference in this picture or in the positions of the eigenvalues pairs. All additional eigenvalues simply joined the cluster at $\lambda=1$.

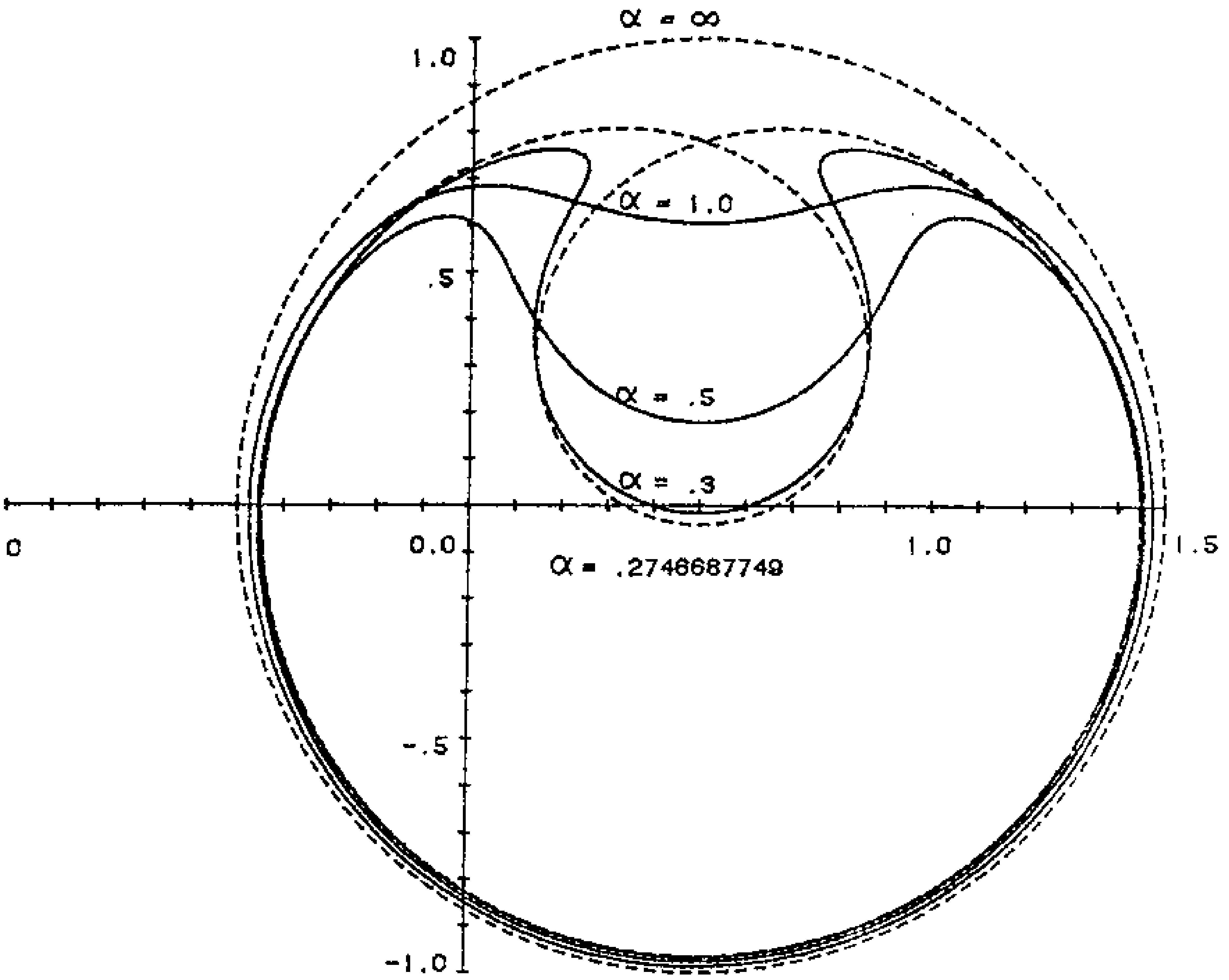

FIG 4. *The curves* $f(x,y,\alpha)\equiv((x-.5)^2+(y-\alpha)^2)(1-(x-.5)^2-y^2)-.1=0$ *for different values of* α.

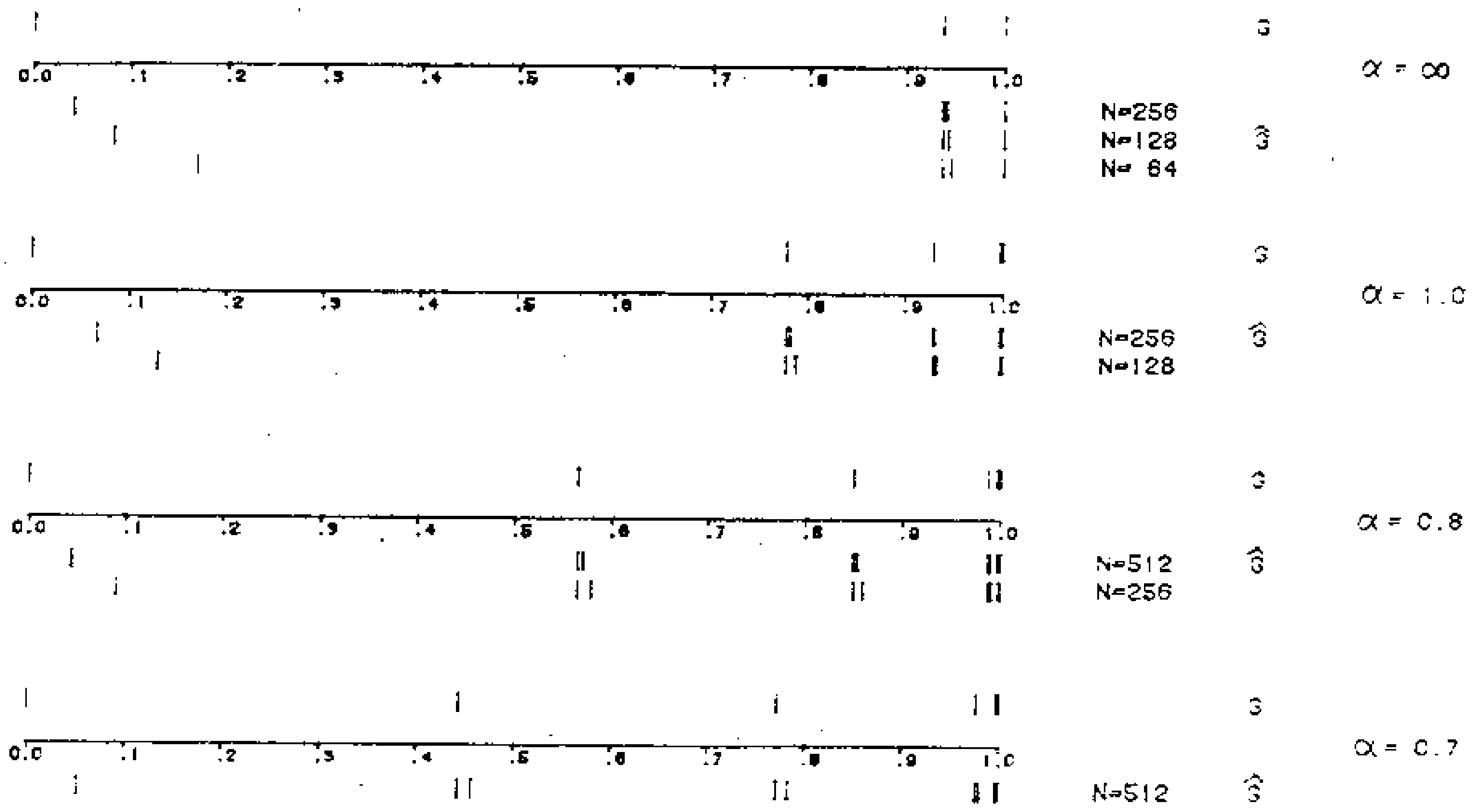

FIG 5. *Eigenvalues of G (virtually independent of N) and of $\hat{G}$ (every second dependent on N) for different values of α in the test case described in the text.*

In the case of extremely low values of N, such as $N=4$ or $N=8$, these eigenvalue properties fail to hold to within truncation accuracy. (In particular, the matrix G is no longer exactly singular.) This indicates that they are not exact properties of the discrete systems. These observations suggest that a smooth curve has some kind of spectrum with the same properties (but with $\lambda=1$ a limit point) and that the discrete method provides exponentially accurate approximations to it for increasing values of N. We have not been able to find any theoretical support for these observations.

The previous discussion has suggested that we can require $t_0=0$ and also consider one equation, for example, the first one, redundant. We write, therefore, (18)

$$(25)\qquad \left[\begin{array}{c:c} g & \hat{g}^T \\ \hdashline \hat{g} & \hat{G} \end{array}\right]\left[\begin{array}{c} t_0 \\ \hdashline \hat{t} \end{array}\right]=\left[\begin{array}{c} b_0 \\ \hdashline \hat{b} \end{array}\right].$$

This leaves us to solve

$$(26)\qquad \hat{G}\hat{t}=\hat{b}.$$

The matrix G is positive semidefinite with only one eigenvalue equal to zero. Therefore, $\hat{G}$ is strictly positive definite, again with a cluster at $\lambda=1$. In the case in Example 1, $\hat{G}$ has one eigenvalue equal to $1/N$ and all the others equal to one. Fig. 5 shows the eigenvalues of $\hat{G}$ corresponding to those of G described earlier in Example 2.

In every case we have studied, we have also noticed the same trend in the eigenvalues of $\hat{G}$. As we mentioned above, G was found to have an eigenvalue $\lambda_1=0$, then double eigenvalues $\lambda_{2,3}$, $\lambda_{4,5}$, etc., clustering at $\lambda=1$. For each double eigenvalue of $G, \hat{G}$ must have a single eigenvalue and the remaining eigenvalues of $\hat{G}$ must lie between the pairs for G. We have noticed that these eigenvalues converge from above to $\lambda_1=0$, to $\lambda_{2,3}$, to $\lambda_{4,5}$, etc., with a rate which seems to be proportional to $1/N$ as N increases.

5. Practical implementation of the mapping method. Conjugate gradient iterations are applied to (26) in a straightforward manner (see for example Luenberger [8] for a description of the method). We simplify the notation of (26) to

$$Hx=b. \tag{27}$$

The method is initialized by the steps

$$\begin{aligned} &\text{choose} \quad x_0=0,\\ &\text{compute} \quad r_0=b-Hx_0,\\ &\text{set} \quad p_0=r_0, \end{aligned} \tag{28}$$

and then iterated for $i=0,1,2,\ldots,$

$$\begin{aligned} a_i&=\frac{r_i^Tr_i}{p_i^THp_i},\\ x_{i+1}&=x_i+a_ip_i,\\ r_{i+1}&=r_i-a_iHp_i,\\ b_i&=\frac{r_{i+1}^Tr_{i+1}}{r_i^Tr_i},\\ p_{i+1}&=r_{i+1}+b_ip_i. \end{aligned} \tag{29}$$

We stop iterating when the changes in the x_i are sufficiently small. The level of errors that we desire to reach depends on the current accuracy in the quadratically convergent outer iteration. The ith approximation in the conjugate gradient method can be shown to minimize

$$\|x-x_i\|_H{}^2=(x-x_i)^TH(x-x_i), \tag{30}$$

over all approximations of the form

$$x_i=x_0+P_{i-1}(H)\cdot H\cdot(x-x_0), \tag{31}$$

where P_{i-1} is any polynomial of degree $i-1$. This result is very favorable in cases with eigenvalue distributions like the one in Fig. 5 but with their exact positions unknown. Complete convergence is assured in the same number of steps as there are distinct groups of eigenvalues. Convergence to sufficient accuracy may take still fewer steps.

The computational cost of the inner iteration is dominated by the fast Fourier transforms. Finding $\mathbf{d}$ (10) costs one transform over N points, or equivalently two transforms over $N/2$ points (denote for simplicity 2 FFTs). Initializing the conjugate gradient method adds 3 FFTs (to find $\mathbf{b}$) and each iteration costs another 4 FFTs (to evaluate Hp_i). When the iterations are finished, 2 FFTs are needed to find $\mathbf{t}_1$. One outer iteration with K inner iterations will therefore cost $7+4K$ FFTs. In the last section, we will see that K is usually about six, which means that each outer iteration typically requires about 30 FFTs (complex transforms over $N/2$ points).

The mapping method requires information about the boundary curve J in two connections:

i) To find the tangent directions at each point.

ii) To move a point back to the curve after it has been moved in the tangential direction.

Any parameter representation of J can be used (for example polar coordinates if the region is "starshaped", spline representation between discrete points, etc.). For the test runs described below, we implemented the method assuming the curve J was given in the form

$$f(x,y,\alpha)=0, \tag{32}$$

with fixed α. (The parameter α allowed us to modify the curve.) Routines were provided for evaluating f, $f_x=\partial f/\partial x$ and $f_y=\partial f/\partial y$ at any point $\zeta_0=x_0+iy_0$ near J. A unit tangential vector at ζ_0 on J is obtained as

$$\frac{(-f_y,f_x)}{(f_x^2+f_y^2)^{\frac{1}{2}}}, \tag{33}$$

and a point ζ_0 off the curve J can, to within sufficient accuracy, be brought back to the curve with one step of a quadratically convergent Newton iteration:

$$\begin{aligned} d&=\frac{f}{(f_x^2+f_y^2)}, \\ x_1&=x_0-d\cdot f_x, \\ y_1&=y_0-d\cdot f_y. \end{aligned} \tag{34}$$

Tests have to be made to determine the number of outer and inner iterations. The rules implemented to obtain an automatic code were as follows:

1. Number of outer iterations. Each time an outer iteration is started, a residual vector is obtained (which later forms the right-hand side of the linear system in the inner iteration). Outer iterations are stopped when the maximal element of this vector has not decreased by more than a factor of 2 since the last iteration. Since the outer iterations are quadratically convergent, an improvement by a factor less than two indicates that the rounding or truncation error level has been reached.

2. Number of inner iterations. The inner iterations use conjugate gradients to approximate the vector $\mathbf{t}_0$; i.e., the distances the even-numbered points (apart from ζ_0 which is held fixed) are to be moved. These iterations were performed until all elements of $\mathbf{t}_0$ had settled to within .001 of the size of the maximal element of $\mathbf{t}_0$. These tests can easily be improved. Since the outer iterations are quadratically convergent, the accuracy in the inner iterations ought to be increased correspondingly. Also, the last outer iteration gives only a small improvement (at most by a factor of two). For each special application, a test should be devised which allows this last iteration to be omitted. For example, in the case of solving a time-dependent problem in a slowly changing geometry, it may be possible to use just one outer iteration for each numerical time step in the main problem.

6. Test results. In this section, the application of the mapping method to the following two one-parameter families of curves is described.

Case 1.

$$f(x,y,\alpha)\equiv\left((x-.5)^2+(y-\alpha)^2\right)\left(1-(x-.5)^2-y^2\right)-.1=0. \tag{35}$$

Fig. 4 showed these curves for some different values of α. The value $\alpha=\infty$ gives a circle with center at (.5,0) and radius 1. The curve ceases to be "starshaped" (i.e., ceases to have a single-valued radius in polar coordinates) at $\alpha\approx.7675275331$ and ceases to be simply connected at $\alpha \approx .2746687749$. The mapping method was employed for values of α down to $\alpha = .5$.

Case 2. (Cassini's oval).

$$f(x,y,\alpha) \equiv \left((x+\alpha)^2+y^2\right)\left((x-\alpha)^2+y^2\right)-1=0. \tag{36}$$

Fig. 6 shows these curves for some values of α ranging from 0 to 1. In the case $\alpha=0$, the curve becomes the unit circle. For $\alpha=1$, it ceases to be simply connected. The mapping function from the unit circle can in this case be found in closed form:

$$\zeta(z)=z\left(\frac{1-\alpha^4}{1-(\alpha z)^2}\right)^{\frac{1}{2}}=(1-\alpha^4)^{\frac{1}{2}}z\sum_{n=0}^{\infty}\frac{(2n)!}{(n!)^2}\left(\frac{z\alpha}{2}\right)^{2n}. \tag{37}$$

Tables 1 and 2 illustrate the performance of the mapping method for these two test cases. Fig. 7 shows some corresponding mappings. For each parameter value, the initial guess on the distribution of the points ζ_i was obtained from the solution for the previous value of α together with two Newton iterations to bring these points to the curve for the new α. If the number of points N was doubled, a fourth-order interpolation was used to find the positions of the new points. In most cases, larger continuation steps in α than those shown would also have worked. At the bottom of Table 1, we give one case in which we tested for the largest possible step. Tables 1 and 2 show the maximal distance any boundary point was moved in each step of the mapping. In these two test cases, an approximate guide for finding the largest allowed continuation step seemed to be that the initial point positions should not be in error by more than about 0.25. If points had to be moved further than that tangentially to the curve in the first outer iteration, they are likely to be returned to the curve in

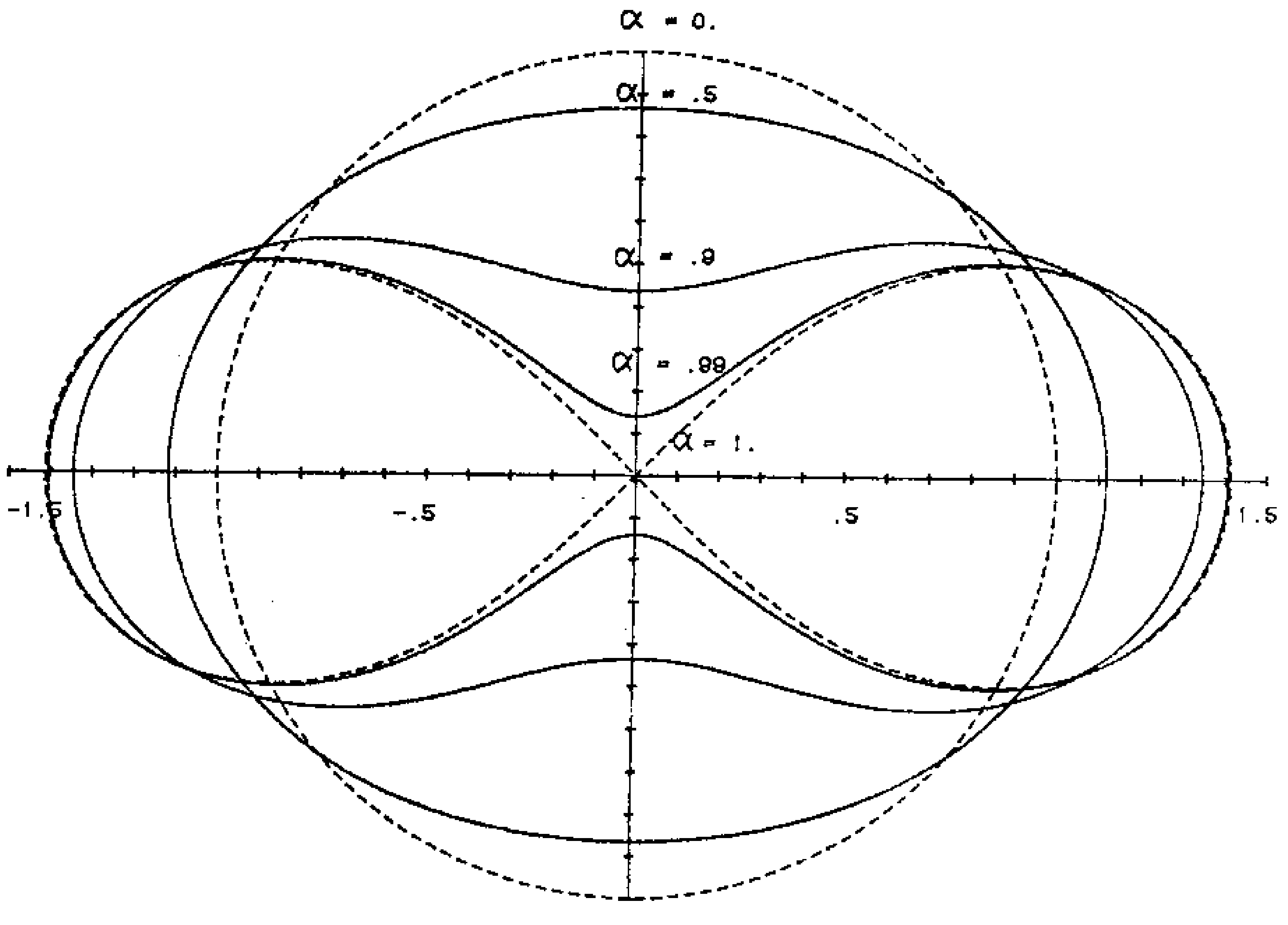

FIG 6. *The curves* $f(x,y,\alpha) \equiv ((x+\alpha)^2+y^2)((x-\alpha)^2+y^2)-1=0$ *for different values of* α.

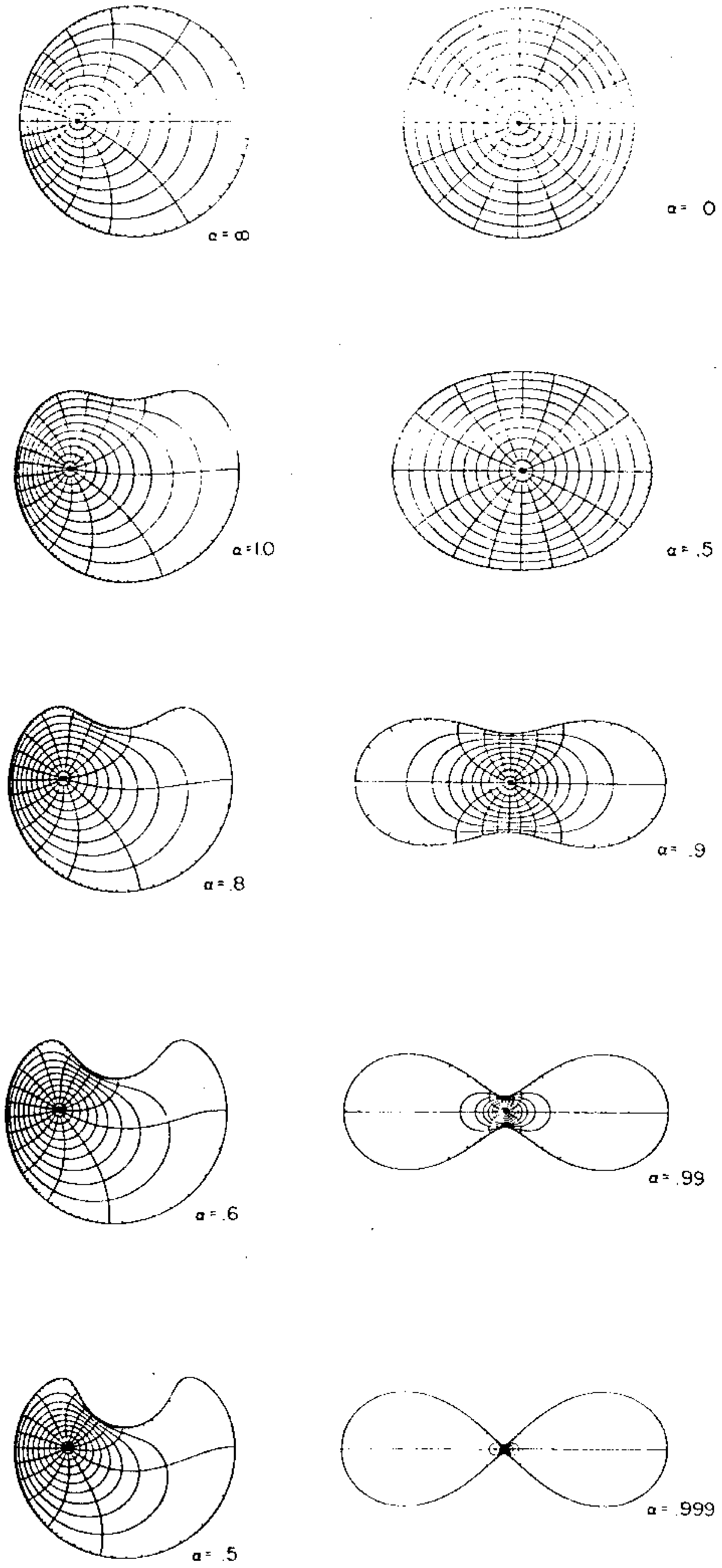

FIG 7. *Examples of conformal mappings in the two test cases.*

places which are no more accurate or may be returned to the curve out of sequential order. If a careful algorithm was developed for returning the point to the curve after each tangential move, still larger continuation steps could probably be used.

We observe in the two test cases that the number of outer iterations seems to depend mainly on the accuracy of the initial guess (which can be read from the column labeled ("Max distance point moved in mapping"). The average number of inner iterations for each outer iteration increased only marginally with increased complexity of the boundary curve. It was found that the method generally produced a uniform absolute accuracy in all the $N/2$ produced Taylor coefficients, and that the error level agreed in size with the first omitted coefficients. The residual vector in the outer iterations also reached this same size. The numbers displayed in the column "Accuracy reached in Taylor coefficients" have been obtained as the maximal element in the final residual vector in the outer iterations. When we keep N fixed and change the parameter α, the changes in the accuracy that was reached only reflect the changes in the decay rates for the leading coefficients.

TABLE 1

Performance of the mapping method in test case 1.

α	N	Nr. of outer iterations	Nr. of inner iterations per outer iterations			Max distance between curves	Max distance point moved in mapping	Accuracy reached in Taylor coefficients	Total comp. time (sec) CDC STAR-100
			Max	Min	Average				
∞	128					Points equidistantly distributed along the curve			
∞	128	7	4	2	3.0	.000	1.000	$.14 \cdot 10^{-13}$	.08
2.0	128	4	4	4	4.0	.047	.028	$.17 \cdot 10^{-11}$	.07
1.5	128	4	5	4	4.3	.088	.058	$14 \cdot 10^{-7}$	.07
1.2	128	4	5	4	4.5	.140	.118	$.33 \cdot 10^{-5}$	.07
1.2	256	2	5	4	4.5	.001	.001	$.97 \cdot 10^{-8}$	.05
1.0	256	4	5	5	5.0	.137	.176	$.17 \cdot 10^{-5}$	.11
1.0	512	2	5	4	4.5	.000	.000	$.40 \cdot 10^{-8}$	.09
.9	512	4	6	5	5.5	.076	.149	$.26 \cdot 10^{-6}$	.19
.9	1024	3	6	5	5.7	.000	.000	$.98 \cdot 10^{-10}$	.28
.8	1024	5	6	6	6.0	.082	.206	$.42 \cdot 10^{-7}$	.48
.8	2048	3	6	4	5.0	.000	.000	$.55 \cdot 10^{-11}$	.53
.75	2048	5	6	3	5.4	.042	.147	$.58 \cdot 10^{-9}$	.94
.72	2048	5	7	4	5.8	.026	.109	$.40 \cdot 10^{-8}$	.97
.70	2048	4	7	4	5.8	.018	.083	$.30 \cdot 10^{-7}$	.78
.70	4096	3	6	4	5.0	.000	.000	$.31 \cdot 10^{-11}$	1.24
.68	4096	5	7	4	5.8	.018	.091	$.38 \cdot 10^{-10}$	2.24
.66	4096	5	7	4	5.8	.018	.101	$.13 \cdot 10^{-9}$	2.24
.64	4096	5	7	4	6.2	.018	.111	$.29 \cdot 10^{-8}$	2.33
.62	4096	4	7	7	7.0	.018	.122	$.19 \cdot 10^{-7}$	2.01
.60	4096	4	8	7	7.5	.018	.135	$.84 \cdot 10^{-7}$	2.10
.60	8192	3	7	5	6.0	.000	.000	$.23 \cdot 10^{-10}$	3.37
.58	8192	5	8	5	6.6	.018	.149	$.78 \cdot 10^{-9}$	5.92
.56	8192	5	8	5	6.8	.018	.165	$.90 \cdot 10^{-9}$	6.05
.54	8192	5	8	5	6.8	.019	.182	$.22 \cdot 10^{-7}$	6.04
.54	16384	3	7	2	4.7	.000	.000	$.28 \cdot 10^{-10}$	6.86
.52	16384	6	7	5	5.7	.019	.200	$.99 \cdot 10^{-9}$	15.20
.50	16384	7	8	5	6.4	.019	.218	$.15 \cdot 10^{-7}$	19.22
						Longest continuation step from $\alpha = .70$, $N = 4096$ that worked			
.64	4096	6	7	7	7.0	.053	.248	$.29 \cdot 10^{-8}$	3.00

TABLE 2
Performance of the mapping method in test case 2.

α	N	Nr. of outer iterations	Nr. of inner iterations per outer iterations			Max distance between curves	Max distance point moved in mapping	Accuracy reached in Taylor coefficients	Total comp. time (sec) CDC STAR-100
0.	128					Points equidistantly distributed. (Exact mapping).			
0.	128	2	3	3	3.0	.000	.000	$.28 \cdot 10^{-13}$	.03
.5	128	5	5	2	2.6	.133	.127	$.14 \cdot 10^{-13}$	.06
.7	128	5	6	2	3.0	.151	.154	$.17 \cdot 10^{-10}$	.06
.8	128	4	3	3	3.0	.114	.139	$.66 \cdot 10^{-7}$	.05
.9	128	4	3	3	3.0	.163	.257	$.81 \cdot 10^{-4}$	.05
.9	256	3	3	3	3.0	.006	.004	$.69 \cdot 10^{-7}$	.06
.93	256	4	3	3	3.0	.068	.143	$.37 \cdot 10^{-5}$	.07
.95	256	3	3	3	3.0	.055	.137	$.47 \cdot 10^{-4}$	.06
.95	512	3	3	3	3.0	.006	.004	$.47 \cdot 10^{-7}$	.10
.97	512	4	3	3	3.0	.069	.212	$.75 \cdot 10^{-5}$	.13
.97	1024	3	3	3	3.0	.004	.002	$.22 \cdot 10^{-8}$	.18
.98	1024	4	3	3	3.0	.044	.172	$.34 \cdot 10^{-6}$	.24
.99	1024	4	4	3	3.5	.058	.291	$.42 \cdot 10^{-4}$	.26
.99	2048	3	4	3	3.7	.012	.010	$.18 \cdot 10^{-6}$	.42
.993	2048	3	4	3	3.7	.023	.154	$.32 \cdot 10^{-5}$	.42
.993	4096	3	4	3	3.7	.004	.003	$.18 \cdot 10^{-8}$	.97
.995	4096	4	4	3	3.5	.018	.146	$.91 \cdot 10^{-7}$	1.26
.997	4096	4	4	3	3.8	.022	.220	$.42 \cdot 10^{-5}$	1.30
.997	8192	3	4	3	3.7	.007	.005	$.65 \cdot 10^{-8}$	2.38
.998	8192	4	4	4	4.0	.014	.176	$.32 \cdot 10^{-6}$	3.29
.999	8192	3	4	4	4.0	.018	.293	$.14 \cdot 10^{-4}$	2.49
.999	16384	2	4	4	4.0	.021	.020	$.16 \cdot 10^{-6}$	3.96

REFERENCES

[1] S. CHAKRAVARTHY AND D. ANDERSON, *Numerical conformal mapping*, Math. Comp., 33(1979), pp. 953–969.

[2] J. W. COOLEY AND J. W. TUKEY, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp., 19(1965), pp. 297–301.

[3] D. GAIER, *Konstruktive Methoden der Konformen Abbildung*, Springer Tracts in Natural Philosophy, 3(1964), Springer, Berlin.

[4] M. H. GUTKNECHT, *Solving Theodorsen's Integral Equation for Conformal Maps with the Fast Fourier Transform, Part I, Theory*, Research Report 79–02, Seminar für angewandte Mathematik, Eidgenössische Technische Hochschule, Zürich, 1978.

[5] ———, *Solving Theodorsen's Integral Equation for Conformal Maps with the Fast Fourier Transform, Part II, Practice*, Research Report 79–04, Seminar für angewandte Mathematik, Eidgenössische Technische Hochschule, Zürich, 1979.

[6] J. K. HAYES, D. K. KAHANER AND R. KELLNER, *An improved method for numerical conformal mapping*, Math. Comp., 26 (1972), pp. 327–334.

[7] P. HENRICI, *Fast Fourier methods in computational complex analysis*, SIAM Rev., 21(1979), pp. 481–527.

[8] D. G. LUENBERGER, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.

[9] R. MENIKOFF AND C. ZEMACH, *Methods for numerical conformal mapping*, J. Comput. Phys., 36(1980), pp. 366–410.

[10] G. T. SYMM, *An integral equation method in conformal mapping*, Numer. Math., 9(1966), pp. 250–258.