JOURNAL OF COMPUTATIONAL PHYSICS **123,** 274–295 (1996)
ARTICLE NO. 0024

The Geometric Solution of Laplace's Equation

EZZAT G. BAKHOUM*

ESD Research Corporation, P.O. Box 2818, Durham, North Carolina 27715

AND

JOHN A. BOARD, JR.

Department of Electrical Engineering, Duke University, Durham, North Carolina 27706

Received June 22, 1994; revised May 30, 1995

A new numerical method for the rapid solution of Laplace's equation in exterior domains and in interior domains with complicated boundaries is presented. The method is based on a formula first stated by J. J. Thomson and later refined by the authors. The mathematical foundations presented allow for the solution of field problems by means of geometric construction principles. Specifically, the method utilizes the concept of representing equipotential surfaces by polynomials for the rapid tracing of these surfaces; and is, therefore, fundamentally different from previously known techniques which are based on discretizing the domain or the boundary of the problem. For the class of problems characterized by irregular domains, the fastest available techniques have traditionally required an $O(M \cdot N)$ computations, where M is the number of points inside the domain at which the solution is computed and N is the number of points used on the boundary. The new method requires an $O(M)$ computations only and is, therefore, more advantageous in large scale calculations. This paper presents only the two-dimensional version of the geometric solution of Laplace's equation.

1. INTRODUCTION

In 1891, J. J. Thomson, the discoverer of the electron, stated a formula that relates the first derivative of the electric field intensity to the mean curvature of an equipotential surface. That formula was later proved by others, but it remained unexploited in any practical purpose to this date.

Over the past three decades, numerical methods have rapidly developed in all areas of physics and engineering. In attempts to solve differential equations with known boundary conditions, especially Laplace's and Poisson's equations, numerical methods have customarily been classified into two broad categories:

- methods that discretize the entire domain, and
- methods that only discretize the boundary.

The FDM (finite difference method) and FEM (finite element method) belong to the first category. The second category includes the MOM (method of moments), but it is typically characterized by the rapidly evolving BEM (boundary element method).

While comparing the advantages and disadvantages of these two categories of algorithms is beyond the scope of this paper (the reader should refer to [22, 23] for that purpose). An important distinction to be noted between the two categories is the inability of algorithms which discretize the entire domain to treat efficiently exterior boundary value problems, such as the problems encountered in aerodynamic calculations, for example, as well as interior problems which have complicated or irregular boundaries. For such types of problems, the boundary element method is almost exclusively applied.

Historically, the BEM have required a number of computations on the order of at least N^2 ($O(N^2)$, for a two-dimensional problem), where N is the number of elements taken on the boundary of the problem. In 1985, Rokhlin [3] presented a rapid algorithm for solving the integral equations which result in the application of the BEM. The Rokhlin method reduced the required number of computations from $O(N^3)$ or $O(N^2)$ to $O(N)$. Unfortunately, however, the method had two main shortcomings:

1. The estimated CPU time of the algorithm was given by $O(N) + \mathrm{O}(M \cdot N)$, where N is the number of points on the boundary and M is the number of points inside the domain at which the solution is to be computed. Rokhlin indicated that his algorithm is the most efficient when the solution has to be computed at a limited number of points inside the domain. However, for substantially large M, the method was much less efficient than other fast solvers.

2. The implementation of the Rokhlin algorithm was quite complicated and challenging for the average physicist

* To the extent that this article may disclose patentable subject matter, the first author reserves those rights.

0021-9991/96 $18.00

or engineer. Usually, the average professional seeks the simplest available tools when the solution of boundary value problems is desired, even if the CPU time requirement is several orders of magnitude higher than other more sophisticated tools. (This, of course, does not apply to commercially available software packages, since the algorithmic complexity in such packages can be transparent to the user.)

An alternative approach to the same problem was also presented by Mayo [5] and still requires an $O(M \cdot N)$ operation. Very recently, Greenbaum *et al.* [4] presented an algorithm that is based on the Rokhlin method for solving Laplace's equation in multiply connected domains.

It is the objective of this paper to provide a new method for solving Laplace's equation in exterior domains and in interior domains with complicated or irregular boundaries, which is:

1. quite simple to implement by the average physicist or engineer; and
2. especially suited for mapping the field inside the entire domain of the problem (where M is large); and
3. quite rapid, even on desktop workstations and personal computers, thereby bringing large scale calculations in complex domains within practical reach.

The new method is an iterative, finite-difference type method, which is based on geometric construction principles. Unlike the presently known finite-difference techniques, the new method does not discretize the domain of the problem into cells, but instead it provides a procedure for computing the shape of the equipotential surfaces directly; where such surfaces are represented by polynomials.

The theoretical and the experimental developments presented here in the following sections show that such a method can be very rapid for quite complex problems. Specifically, it will be demonstrated that the new method requires an $O(M)$ computations only, where M is the number of points inside the domain at which the solution is computed and is, therefore, an optimal order method.

The geometric method is based on Thomson's formula which was discovered more than a century ago, but the potential of which regrettably remained unrecognized. Thomson's formula appeared in a footnote in his edited version of Maxwell's *Treatise* [12] and states that near an equipotential surface, the normal derivative of the magnitude of the electric field E satisfies

$$\frac{\partial \|E\|}{\partial n} = -\|E\| \left(\frac{1}{R_1} + \frac{1}{R_2} \right), \tag{1}$$

where R_1 and R_2 are the principal radii of curvature of the surface at the point under consideration, and n is the normal direction to the surface at the same point.

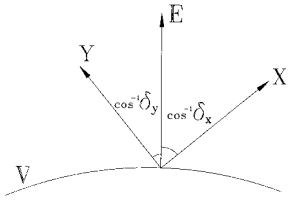

FIG. 1. The electric field vector E is normal to an equipotential curve V at all points. In an arbitrary coordinate system $X - Y$, δ_x, δ_y are the direction cosines of the vector E.

Thomson's formula was later proved by others [14–16]. Recently, the authors [1] presented a generalized version of Thomson's formula that is valid in any direction within the field, not necessarily the direction normal to an equipotential surface, thereby effectively obtaining a formulation for the components of Laplace's equation.

In two dimensions, that general formula is [1]

$$\frac{\partial \|E\|}{\partial y} = -\|E\| \frac{\partial \delta_x / \partial x}{\delta_y}, \tag{2}$$

where x,y are the principal directions, and where δ_x,δ_y are the components of a unit vector tangent to the field line along the principal directions (i.e., δ_x,δ_y are the direction cosines of the electric field vector E; see Fig. 1).

This paper uses Thomson's formula (1) and its generalized version (2) to establish a purely geometric method for the solution of Laplace's equation using both Dirichlet and Neumann boundary conditions. For the most common case, where only the Dirichlet conditions are known, the new geometric method is combined with an iterative technique and may be, therefore, called the "geometric relaxation" method.

Like all relaxation methods, the number of iterations required for convergence of the new method is a function of the initial guess. In Section 2.3, it is shown that the number of iterations is an exponentially decreasing function of the initial guess. It is demonstrated that if a modest initial guess is first obtained with a classical integral-equation solver, by using a small number of points on the boundary, the speed of convergence of the geometric relaxation is substantial and can exceed the speed of the fastest known algorithms available today.

This paper is organized as follows: Section 2.1 introduces the basic concept behind the geometric solution of Laplace's equation, subject to both Dirichlet and Neumann boundary conditions. Section 2.2 presents necessary criteria for reducing the computational errors associated with the numerical algorithm. Section 2.3 describes a geometric relaxation algorithm to encompass the category of practical problems in which only the Dirichlet conditions are known and lists some derivations related to the estimation of per-

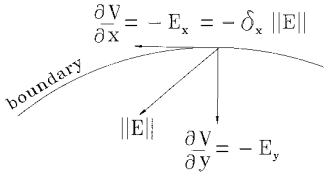

FIG. 2. The potential gradient $\partial V/\partial y$ is known on the boundary, as well as the potential V. This allows the computation of the electric field intensity and direction at every point on the boundary.

formance of such relaxation algorithm. Section 3 discusses some solution examples, in terms of the numerical results obtained from the computer and in terms of some general mathematical and physical insights provided by the new method. Section 4 contains our conclusions.

Appendices A and B contain mathematical derivations which are necessary for understanding the geometric solution and are referred to in the paper where appropriate. Appendix C lists a detailed, step by step description of the practical algorithm which implements the geometric solution inside irregular domains.

2. THEORETICAL FOUNDATIONS

2.1. The Basic Concept

The geometric method will be explained here in the embodiment of interior domains with closed boundaries. The applicability of the method to exterior domains will be immediately apparent once the basic concept is understood.

For the solution of Laplace's equation, a charge-free region must be considered. Inside such a region, a closed equipotential surface cannot exist, since the existence of such surface necessitates the presence of a source or sink within the volume of space enclosed by that surface (for proof, see Appendix A). The boundary of the region where Laplace's equation holds, therefore, must always intersect and never enclose equipotential surfaces.

We shall now assume that both Dirichlet and Neumann boundary conditions are given, i.e., both the potential and the normal potential gradient. In Section 2.3 we shall relax the Neumann requirement, since, in most cases, the potential gradient on the boundary is not known *a priori*.

The electric field intensity and direction at every point on the boundary (Fig. 2) can be found from

$$\begin{aligned} \|E\| &= \sqrt{(\partial V/\partial x)^2 + (\partial V/\partial y)^2} \\ \delta_x &= \frac{-\partial V/\partial x}{\|E\|}, \quad \delta_y = \frac{-\partial V/\partial y}{\|E\|}. \end{aligned} \tag{3}$$

Let the potential distribution along the boundary be labeled V_1, V_2, ..., V_j, from the highest to the lowest, as shown in Fig. 3.

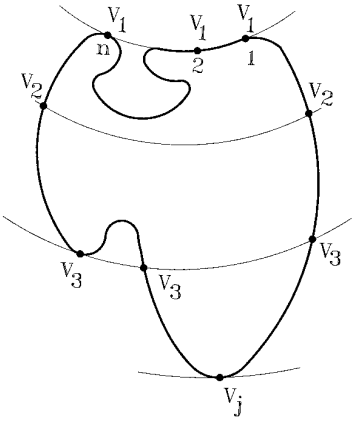

FIG. 3. The boundary of a region where Laplace's equation holds. The boundary intersects equipotential surfaces labeled V_1, V_2, ..., V_j, from the highest to the lowest.

By using a linearized Taylor series at every point on the equipotential curve V_2, we have

$$V_2 \approx V_1 + \frac{\partial V_1}{\partial s}\Delta s, \tag{4}$$

where s is the distance along the electric field line (Fig. 4). But since

$$\frac{\partial V}{\partial s} = -\|E\|, \tag{5}$$

where the potential is decreasing, thus, in Fig. 4,

$$\begin{aligned} \Delta s_{11} &= \frac{V_1 - V_2}{\|E_{11}\|}, \\ \Delta s_{12} &= \frac{V_1 - V_2}{\|E_{12}\|}, \ldots, \Delta s_{1n} = \frac{V_1 - V_2}{\|E_{1n}\|}. \end{aligned} \tag{6}$$

Now, given n support points on the equipotential V_2,

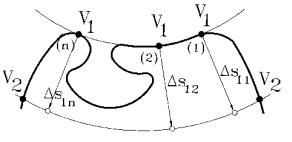

FIG. 4. Equipotential surface V_2 can be found by calculating several distances Δs_{11}, Δs_{12}, ..., Δs_{1n} along the electric flux lines propagating from surface V_1 towards V_2. To a first-order approximation, such distances could be calculated as the ratio of potential difference over the electric field intensity.

plus two points on the boundary, a polynomial of degree $n + 1$ can be fitted to represent the equipotential curve V_2.

To proceed further inside the domain, we must now obtain the electric field intensity at n points on the equipotential V_2 in order to compute a new set of distances $\Delta s_{2i} = (V_2 - V_3)/\|E_{2i}\|$, and hence determine the location of the equipotential curve V_3.

Again, a linearized approximation

$$\|E_{2i}\| \approx \|E_{1i}\| + \frac{\partial \|E_{1i}\|}{\partial s} \Delta s_{1i}, \qquad i = 1, ..., n, \tag{7}$$

can be used. The quantity $\partial\|E_{1i}\|/\partial s$ can be found from one of two possible formulas:

• if the curvature of the equipotential V_1 is known at point i, $i = 1, ..., n$ (see Fig. 4), then Thomson's formula in two dimensions,

$$\frac{\partial \|E_{1i}\|}{\partial s} = -K_{1i}\|E_{1i}\|, \tag{8}$$

can be used directly, where $K = 1/R$ is the curvature. In this case,

$$\|E_{2i}\| \approx (1 - K_{1i}\Delta s_{1i})\|E_{1i}\|; \tag{9}$$

• if the curvature is unknown at point i, a variation of the general formula (2) must be used. This variation can be derived as follows: first, we observe that

$$\frac{\partial \|E\|}{\partial s} = \frac{\partial \|E\|}{\partial x}\delta_x + \frac{\partial \|E\|}{\partial y}\delta_y. \tag{10}$$

By selecting a coordinate system such as shown in Fig. 2 and substituting from the general formula (2) into Eq. (10), we get

$$\frac{\partial \|E\|}{\partial s} = \frac{\partial \|E\|}{\partial x}\delta_x - \frac{\partial \delta_x}{\partial x}\|E\|. \tag{11}$$

This result shows that the first derivative of the electric field intensity along the flux line can be completely determined by measurements taken along the boundary.

In order to compute the electric field intensity along the points of equipotential curve V_3, the use of Thomson's formula is now required at the n points on equipotential V_2 which lie inside the boundary, since the curvature K can be computed at those points. For the two points on the boundary, formula (11) must be used. In general, the quantities $\partial\|E\|/\partial x$, $\partial\delta_x/\partial x$ must be evaluated along the entire boundary before the solution can be carried out, since the curvature of equipotentials is not assumed to be known at the boundary points.

Inside the domain, the curvature of an equipotential at any point can be computed from the polynomial representation $y = f(x)$ of the equipotential by using the formula [7, 8]

$$K = \frac{y''}{(1 + y'^2)^{3/2}}, \tag{12}$$

where $y' = f'(x)$ and $y'' = f''(x)$.

Since the first and second derivatives of the function $y = f(x)$ are required at the support points, it is most convenient to use the method of interpolation by cubic splines [8] to represent that function, since the method of interpolation by cubic splines features some readily available formulas for computing such derivatives. In Appendix B, it is shown how each equipotential curve can be represented as a group of cubic splines and how the curvature at the support points can be computed.

Remark 2.1. From Appendix B and from the description given above, it is apparent that one matrix operation (basically, solution of a tri-diagonal system) is required for each equipotential curve obtained. The number of unknowns in each matrix operation is equal to the number of support points on the curve. As can be observed from Appendix B, the solution of such a tri-diagonal system requires a number of computations equal to the number of unknowns. Therefore, for a total number of M support points obtained inside the domain, the solution requires an $O(M)$ computations.

2.2. Criteria for Selecting the Step Size ($V_r - V_{r+1}$)

In computing the incremental distance Δs to a first-order approximation according to Eq. (6), we must find proper criteria for selecting the step size of potential $V_1 - V_2$, $V_2 - V_3$, etc. A too large value will render the first-order approximation invalid, and a too small value may unnecessarily increase the computation time. Such criteria will now be found.

In the Taylor expansion of Eq. (4), the correct formulation for the incremental distance Δs is

$$\Delta s = \frac{V_1 - V_2}{\|E\|} + \frac{1}{\|E\|} \times \left(\frac{\partial^2 V_1/\partial s^2}{2!}(\Delta s)^2 + \frac{\partial^3 V_1/\partial s^3}{3!}(\Delta s)^3 + ... \right), \tag{13}$$

but we note that

$$\frac{\partial^2 V}{\partial s^2} = -\frac{\partial \|E\|}{\partial s} = K\|E\|; \tag{14}$$

therefore

$$\frac{1}{\|E\|}\cdot\frac{\partial^2 V/\partial s^2}{2!}(\Delta s)^2=\frac{K}{2}(\Delta s)^2. \tag{15}$$

An estimation for the lower bound of the error in the computed distance Δs can now be found as follows: by neglecting terms of third order and higher in Eq. (13), we have, by substitution from (15),

$$\Delta s\approx\frac{\Delta V}{\|E\|}+\frac{K}{2}(\Delta s)^2, \tag{16}$$

where the variable ΔV represents $V_1 - V_2$, or $V_2 - V_3$, etc., where $V_r - V_{r+1}, r = 1, ..., j$, is the step size of potential.

From Eq. (16),

$$\Delta s\approx\frac{1\pm\sqrt{1-4\cdot(K/2)\cdot(\Delta V/\|E\|)}}{K} \tag{17}$$

from which, when

$$\frac{2K\,\Delta V}{\|E\|}=1,$$

we get

$$\Delta s=\frac{1}{K}=\frac{2\,\Delta V}{\|E\|} \tag{18}$$
$$= \text{radius of curvature of the equipotential curve.}$$

Therefore, it is clear that when the step size $\Delta V = \|E\|/2K$, the improved estimation for Δs is $2\,\Delta V/\|E\|$, while the original estimation was $\Delta V/\|E\|$. An error of about the same order of magnitude as the original estimation is a large error.

Accordingly, it will be apparent that, given specific values of $\|E\|$ and K at any point on the equipotential curve, a step size ΔV must be selected such that

$$\Delta V\ll\frac{\|E\|}{2K}. \tag{19}$$

(Since ΔV must be unique for all points on the curve, the best accuracy is obtained by selecting the minimum value of $\|E\|/2K$ on the curve.)

It should be further noted from Eq. (15) that the error, represented by the second and higher derivatives, is directly proportional to the curvature of the equipotential at any point. Accordingly, from Eq. (19) it can be seen that a larger step size ΔV can be taken if the electric field intensity is higher, for some value of K. Equation (19), therefore, allows for the implementation of an ''adaptive'' algorithm. At regions where the electric field intensity $\|E\|$ is large compared to the value of the curvature K, a large step size ΔV can be selected, thus reducing the overall computation time without loss of accuracy.

When $K = 0$ (the case of a flat equipotential), the second derivative in the Taylor expansion of Eq. (13) vanishes. The third derivative contains $\partial K/\partial s$. It can be easily seen, by using the relation $\partial\|E\|/\partial s = -K\|E\|$, that $\partial K/\partial s|_{K=0} = -(\partial^2\|E\|/\partial s^2)/\|E\|$; however, since both $\partial K/\partial s$ and $\partial^2\|E\|/\partial s^2$ are unknowns, it is not generally possible to find a condition similar to (19) when $K = 0$. In this case, the error in the expansion can be minimized by selecting a small displacement Δs, i.e., selecting $\Delta V \ll \|E\|$.

2.3. A Geometric Relaxation Algorithm

In the preceding analytical treatment, a geometric method for solving Laplace's equation was introduced. The method is based on the assumption that the distributions of both the potential and the normal potential gradient are known along the boundary. Since, for most practical cases, the normal potential gradient is unknown, the boundary conditions must first be initialized before the geometric solution can proceed.

The algorithm now proposed consists of the following two steps:

1. First, initialize the boundary conditions by using the classical integral-equation method (see, for example, Brebbia [17]), with a small number of points being taken on the boundary to obtain a reasonable guess for the potential gradient.

2. Apply the geometric method for solving Laplace's equation, using the initial guess obtained as a ''seed'' to start the solution.

The rest of this paper is concerned with the development of an iterative scheme to improve such solution and with providing some theoretical predictions for the performance of such iterative scheme. As we shall demonstrate, such a technique still preserves the $O(M)$ computational effort of the geometric method and will, therefore, be significantly faster than the classical $O(M \cdot N)$ techniques.

2.3.1. *Qualitative Description of the Algorithm*

The geometric relaxation algorithm will be explained here, again, within the embodiment of interior domains with closed boundaries. The applicability of the algorithm to exterior domains will be immediately apparent once the basic concept is understood. Solution examples for each of the cases will be presented in Section 3. Appendix C lists a detailed, step by step description of the practical implementation of this algorithm.

First, the initial guess of the Neumann conditions along the boundary must be obtained as outlined above.
The points of highest and equal potential on the boundary

are then selected, following the scheme of Section 2.1. By building the equipotential surfaces successively, as outlined in Section 2.1, the region of the problem will be eventually filled to completion. This will be considered as the first "iteration" in the geometric relaxation method. The method initially requires, as stated, the distributions of potential and electric field intensity and direction along the boundary. Since the potential distribution on the boundary is given *a priori*, then the electric field intensity and direction along the boundary must now be modified in order to start a new iteration. If the electric field intensity at a particular point is modified, the new direction can be found from Eq. (3), as $\delta_x(\text{new}) = -(\partial V/\partial x)/\|E_{\text{new}}\|$. The geometric relaxation algorithm, therefore, consists of the steps of starting with a poor approximation of the electric field intensity (or potential gradient) on the boundary, carrying the geometric solution of Laplace's equation inside the domain, and then improving the original estimation of gradient and repeating the geometric solution procedure.

The manner by which the intensity of the electric field on the boundary can be modified at each new iteration will be now explained, together with the conditions necessary for convergence of the algorithm.

2.3.2. *Conditions of Convergence*

LEMMA. *The number of iterations required for convergence of the geometric relaxation algorithm is given by*

$$i = \frac{\log[\log n.e._i/\log n.e._1]}{\log 2},$$

where $n.e._1$ is the maximum normalized error in the initial Neumann conditions on the boundary of the problem, and $n.e._i$ is the normalized error after the i^{th} iteration.

Remark 2.2. The value of the normalized error at any point on the boundary ranges from 0 to 1; i.e., $0 < n.e._1 < 1$ along the boundary. If such normalized error is equal to 1 at any point on the boundary, the initial value of potential gradient at that point is in error by an amount equal to 100% of the actual value. In other words, the normalized error is the ratio of the absolute error over the exact value. After i iterations, the normalized error at the same point is reduced to $n.e._i$. If, for instance, $n.e._i = 10^{-7}$, and the exact value of gradient equals 100, for example, then the solution has converged to five digits. But if the exact value is 10, then the solution has converged to six digits, etc.

Proof. In implementing the geometric relaxation algorithm, we must modify the value of the field intensity $\|E\|$ on the boundary at the starting point of each field line before carrying a new iteration inside the domain. Let, at a given point, the value of $\|E\|$ be modified by a factor α_i at each new iteration, i.e.,

$$\|E_{i+1}\| = \alpha_i \|\|E_i\|; \tag{20}$$

thus,

$$\begin{aligned} \|E_2\| &= \alpha_1 \|E_1\|, \\ \|E_3\| &= \alpha_2 \|E_2\| = \alpha_2 \alpha_1 \|E_1\|, \\ \|E_i\| &= \alpha_{i-1} \ldots \alpha_2 \alpha_1 \|E_1\|. \end{aligned} \tag{21}$$

In order to determine how the factor α_i is calculated at each new iteration, we first set the following logical rule for adjusting the field intensity $\|E_i\|$:

$$\|E_{i+1}\| = \|E_i\| \pm \|E_i\| \, n.e._i, \tag{22}$$

where $n.e._i$ is the normalized error at the ith iteration step. (Note that $n.e._i \to 0$ as the solution proceeds, since the error is being reduced.) Thus,

$$\alpha_i = 1 \pm n.e._i. \tag{23}$$

Let $\|E_i\|$ at any iteration step be less than $\|E_{\text{exact}}\|$, where $\|E_{\text{exact}}\|$ is the exact value of the field intensity, so the sign in Eq. (23) is positive. Now, the normalized error after the first iteration will be given by

$$n.e._2 = \frac{\|E_{\text{exact}}\| - \|E_2\|}{\|E_{\text{exact}}\|} = 1 - \frac{\alpha_1 \|E_1\|}{\|E_{\text{exact}}\|}. \tag{24}$$

But

$$n.e._1 = \frac{\|E_{\text{exact}}\| - \|E_1\|}{\|E_{\text{exact}}\|} = 1 - \frac{\|E_1\|}{\|E_{\text{exact}}\|}; \tag{25}$$

therefore,

$$\begin{aligned} n.e._2 &= 1 - \alpha_1(1 - n.e._1) \\ &= 1 - (1 + n.e._1)(1 - n.e._1) \\ &= (n.e._1)^2. \end{aligned} \tag{26}$$

Similarly, it can be shown that

$$\begin{aligned} n.e._3 &= (n.e._2)^2 = (n.e._1)^4, \\ n.e._4 &= (n.e._1)^8, \\ &\text{etc.}, \end{aligned} \tag{27}$$

where $n.e._i$ is the normalized error before the i^{th} iteration step. Therefore,

$$n.e._{i+1} = (n.e._1)^{2^i}. \tag{28}$$

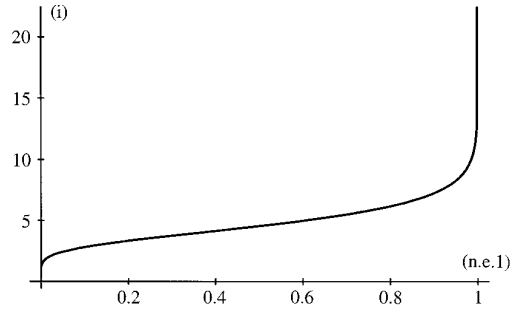

FIG. 5. A plot of the number of iterations (i) versus the normalized error ($n.e._1$).

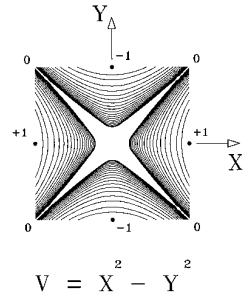

FIG. 6. The geometric solution of the analytic function $V = X^2 - Y^2$, which satisfies Laplace's equation.

By taking the logarithm twice for both sides of the equation, it can be easily verified that

$$i = \frac{\log[\log n.e._i / \log n.e._1]}{\log 2}, \tag{29}$$

where $n.e._i$ has been substituted for $n.e._{i+1}$ for simplicity. (Note that $n.e._i$ represents the normalized error after i iteration steps have been completed.) ▌

A plot of the function i of Eq. (29) is shown in Fig. 5 for $n.e._i = 10^{-7}$ (constant) and for $0 \leq n.e._1 \leq 1$.

It is clear that an infinite number of iterations is required for convergence when the initial normalized error $n.e._1$ approaches unity. However, it can also be seen that the function of Eq. (29) is very well behaved for values of error up to 80%. This suggests that if a modest initial guess of only 20% accuracy is used to initialize the geometric relaxation algorithm, the operation count of the algorithm is expected to be very low. Of course, such modest guess can be obtained with a standard integral-equation solver and with a small number of points being taken on the boundary.

Remark 2.3. The infinite number of iterations required for convergence when the initial normalized error approaches unity is a result which will be further understood from the description of the algorithm given in Appendix C, as a direct consequence of the dependence of the geometric method on a sensitive point on the boundary called the ''inflection'' point, at which the sign of the normal potential gradient flips from positive to negative or vice-versa.

Having obtained the above result, we shall now ask the following question: what is the total computational effort of the proposed geometric relaxation algorithm? As we have demonstrated in Section 2.1, the basic geometric method takes an $O(M)$ computations to obtain the solution at M points inside the domain. When the method is repeated i times, the total computational effort is of $O(i \cdot M)$. However, the graph of Fig. 5 above predicts that, with a modest initial guess on the boundary, the number of iterations i will be less than 5. With such a small number of iterations, therefore, the method essentially requires an $O(M)$ computations. It should be noted, of course, that such a result is independent of the number of points N initially taken to initialize the boundary conditions (this will be further clarified in Section 3).

We have tested a variety of problems and found that by taking a small number of points on the boundary, e.g., 32–64 points, and using a standard integral-equation solver, the accuracy of the estimated potential gradient along the boundary is much higher than 20% and is generally higher than 90% (this corresponds to $n.e._1 = 0.1$). Such results also seem to agree with other results published independently in several references [3, 4, 17].

3. SOLUTION EXAMPLES AND PRACTICAL CONSIDERATIONS

In this section we shall give solution examples of some classical boundary value problems, using both Dirichlet and Neumann boundary conditions, or the Dirichlet conditions only, and we shall describe some practical and physical considerations that should be observed when the method is applied.

3.1. Problems Subject to Both Dirichlet and Neumann Boundary Conditions

A program was written to test the basic geometric method for solving Laplace's equation, as described in Section 2.1. The inputs to the program are the distributions of the potential and the normal potential gradient along the boundary of the problem.

3.1.1. *Interior Problem*

Consider, first, a potential function which satisfies Laplace's equation, and which can be calculated analytically. An example of such a function is $V = X^2 - Y^2$. For this function, the potential $V = +1$ at $X = \pm 1$, $Y = 0$, and $V = -1$ at $X = 0$, $Y = \pm 1$, as can be seen in Fig. 6. Further, $V = 0$ at the edges of the square region,

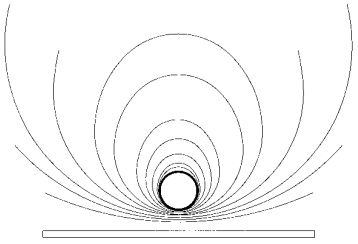

FIG. 7. The geometric solution of an exterior potential problem.

TABLE I

No. of points	Initialization	Geometric relaxation	No. of iterations	Total time
32	7.1	41.5	5	48.6
40	10.5	31.2	4	41.7
48	16.9	25.1	3	42.0
64	27.6	25.3	3	52.9

Note. Times are in seconds.

which suggests the presence of two intersecting, planar equipotentials at the center. The program was made adaptive according to the criteria described in Section 2.2; i.e., it computes the value of $\|E\|/2K$ at all points on each new equipotential curve and selects the minimum. Such a value is then divided by some constant and taken as the step size ΔV for computing the distances Δs along the flux lines. The program used the improved, second-order approximation of Δs, given by Eq. (17), instead of the first-order approximation. The result is shown in Fig. 6.

It should be noted that each equipotential curve in the solution is actually a group of cubic splines, each given by Eq. (32) (Appendix B), and each defined between two points on the curve.

In this particular example, the accuracy of the computed equipotentials can be found as follows: at a given point X,Y on an equipotential curve, the exact value of potential is $V = X^2 - Y^2$. However, the computed value of the potential for such a curve will be different. The error will then be the difference between the exact and computed values of potential. The maximum error within the domain of the problem can then be found. It was found that the error, in fact, decreases when the step size ΔV is made much smaller than the minimum value of $\|E\|/2K$ on the curve, and it increases when the step size increases. By properly selecting ΔV, it was found that the solution can be accurate to seven digits or more (i.e., the magnitude of the error can be less than 10^{-7}). Note that such accuracy is a feature of the method of interpolation by cubic splines and may not be generally obtainable by other interpolation schemes (see Ref. [8]).

Remark 3.1. A SUN Sparcstation 2 was used to solve the examples presented in this section. To calculate the potential accurately to two digits at all points inside the domain, the execution time on the CPU was negligible. However, the execution time was found to grow linearly for higher accuracies, confirming that the method is an $O(M)$ method.

3.1.2. *Exterior Problem*

Figure 7 shows the solution obtained by the program for another classical example: the Van-de-Graaf generator. In this problem, a metallic sphere is charged to a potential of 100 V and raised to a certain distance above the ground, which is held at a potential of 0 V. By knowledge of the boundary conditions at the sphere, the program traced the equipotentials shown in the figure. For this problem, the Neumann conditions were first obtained by using a standard integral-equation solver.

3.2. Problems Subject to Dirichlet Conditions Only

An algorithm that implements the geometric solution by relaxation, as described in Section 2.3 and more fully in Appendix C, was written and tested. The test cases shown below are again fully described in Appendix C.

3.2.1. *Interior Problem*

EXAMPLE 1. Consider again the problem $V = X^2 - Y^2$ shown in Fig. 6 above. For this problem, a classical $O(N^2)$ integral-equation solver has been used to obtain the initial guess of potential gradient; and the number of points used on the boundary varied from 32 to 64 points (8 to 16 points on each side of the square). At 32 points, the classical solver took 7.1 s to obtain the boundary conditions, and then the geometric relaxation algorithm took 41.5 s and five iterations to converge to seven digits (i.e., to reduce the absolute error to 10^{-7}). At 64 points, the classical solver took 27.6 s to execute, and then the geometric relaxation algorithm took 25.3 s and three iterations to converge to seven digits. These results are summarized in Table I.

It should be noted from the table that the total execution time reached a minimum of 41.7 s at 40 points. For higher and lower selections of the number of points on the boundary, the execution time was higher. Therefore, by initializing the boundary conditions at 40 points on the boundary, the total execution time for this problem was minimized. Of course, such optimal selection of the number of points depends on the problem and cannot be determined with certainty.

The total number of points inside the domain, M, at which the solution was computed in this problem was about 62,000 points. (Such a large number of points was necessary to obtain the accuracy of 10^{-7}).

Remark 3.2. It should be noted here that a number of points on the boundary much higher than the initial number N is actually taken as the geometric solution proceeds. (Of course, we cannot obtain an accuracy of 10^{-7} by using 32 points on the boundary without violating the sampling theorem!). In practice, as the algorithm traces equipotentials inside the domain, it may require to select a point on the boundary at which the potential V corresponds to a newly computed equipotential curve (see Fig. 4), but such a point may not belong to the set of discrete points already available on the boundary. In such case, the potential gradient at that point on the boundary is obtained by linearly interpolating between the two discrete points surrounding that point (more in Appendix C). Of course, as outlined above, the geometric relaxation algorithm can quickly improve the estimation of potential gradient within few iterations. Clearly, this scheme will always require an $O(M)$ operations regardless of the number of points taken on the boundary.

After the results described above were obtained, the classical integral-equation solver was then applied to the problem. To reduce the error to 10^{-7}, the program took about 2000 points on the boundary and 32 h of CPU time! Finally, the faster Rokhlin algorithm was applied to this problem. The procedure which computes the Neumann conditions on the boundary was considerably faster than in the classical case; however, the timing for the procedure which calculates the potential inside the domain (at 62,000 points) was approximately 1.2 h. Such timing was expected, since, as outlined earlier, such a procedure requires an $O(M \cdot N)$ operations, where $M = 62{,}000$ and $N = 2000$ in this example. Comparing with the worst execution time of 41.5 s, taken by the geometric relaxation algorithm, we see that a speedup factor of at least 100 can be obtained with the geometric method.

EXAMPLE 2. One of the important applications in biomedical research is the tracing of potential fields inside a human heart which is being exposed to a defibrillation shock. In this application, recording electrodes are placed at several locations on the surface of the heart, and the potential at these locations is measured. (See Ref. [6].) Conventional Laplace solvers are then used to trace the equipotential surfaces inside the heart.

While this problem is a complex three-dimensional problem, Fig. 8 shows a typical two-dimensional representation of the right posterolateral view of the heart. The figure also shows an arbitrary distribution of potentials on the boundary (in volts), and the equipotentials obtained by the geometric relaxation algorithm for the given distribution of potentials. The number of points inside the domain at which the solution was computed in this problem was about 10,000 points.

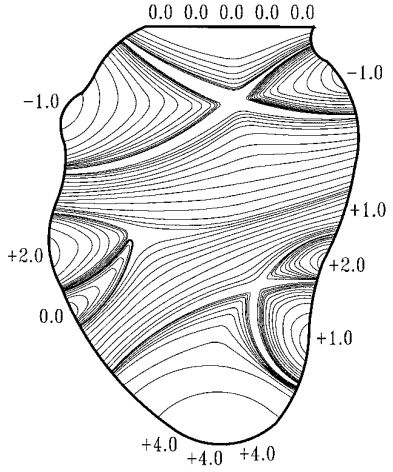

FIG. 8. Interior Dirichlet problem.

3.2.2. *Exterior Problem*

Consider the problem of the metallic sphere raised above the ground shown in Fig. 7. For this problem, the potential along the field lines which end on the flat surface tends to a constant (here V = 0 at the surface), while the potential along the lines ending at infinity (the far field potential) obeys the logarithmic decay rule.

Note that in this problem, points on both the plane and the sphere must be taken in order to build the interaction list which is necessary for calculating the initial value of potential gradient on the points of the sphere. However, when the geometric relaxation is applied, the solution proceeds at the surface of the sphere only. The far-field and the near-field conditions are then applied at each field line separately.

3.3. Practical and Physical Considerations

From the examples shown in Figs. 6 and 8 above, a remarkable characteristic of the geometric solution will be apparent; as observed, the algorithm cannot proceed in regions where very high curvatures exist. As can be seen in Fig. 6, the program could not trace equipotentials near the center of the square where $K = \infty$. In Fig. 8, K reaches infinity at three different locations inside the domain, and the program cannot proceed near such locations. This strange phenomenon has also been observed in several other test cases. In practice, as $K \to \infty$ (a point of singularity), the selected step size $\Delta V \to 0$ in order to minimize the error, according to the criteria outlined in Section 2.2 above. Accordingly, the computed distance $\Delta s \to 0$.

This shortcoming, however, is not a deficiency of the geometric method. By observing that Thomson's formula is a first-order differential equation, it can be easily verified that the solution of such a formula is $\|E\| = \|E_0\| \exp(-Ks)$,

where $\|E_0\|$ is the field intensity at the equipotential surface and s is the distance along the field line. Now, by observing that $\|E\| = |\partial V/\partial s|$, it can be seen that when $K \to \infty$, $|\partial V/\partial s| \to 0$; i.e., in the neighborhood of $K = \infty$ the potential remains essentially constant. This means that the difference in potential between two successive equipotential curves in such a region can be practically neglected. In practice, computer output showed that the program may stop at an equipotential curve at which the potential is, for example, $V = 10^{-4}$ V, while the potential at the singular point is $V = 0$ V. For the example of Fig. 6, this conclusion means that the potential throughout the blank region at the center of the square is nearly constant and equal to 0.

Now, we shall raise the following question: if the geometric solution cannot proceed in regions where $K \to \infty$, what physical implications arise from this phenomenon and how can the program be practically implemented? In practice, if the adaptive algorithm cannot proceed past a particular equipotential surface, this is due to the quantity $\|E\|/2K$ being very small at one or more points on that surface. As discussed, the smaller this quantity becomes at any point, the higher the error in the Taylor expansion in the neighborhood of that point and, hence, the smaller the displacement Δs which must be taken in order to suppress the error terms. This is one important physical aspect in the solution of Laplace's equation which appears clearly in the present method, but which does not show in any other numerical technique. Apparently, this relationship between the error and the curvature of the equipotentials was unknown to other researchers in the field of numerical analysis.

Remark 3.3. It can be easily observed that, in general, when potential problems are solved numerically, the error always increases in the neighborhood of $K \to \infty$, regardless of the numerical method being used. This can be observed from Thomson's formula (Eq. (8)), and by observing that $\partial\|E\|/\partial s = -\partial^2 V/\partial s^2$; then if $K \to \infty$, $\partial^2 V/\partial s^2 \to \infty$. It can also be verified that the higher derivatives of V tend to infinity. But since any numerical method for solving Laplace's equation must neglect some high-order terms in the Taylor expansion of the potential V at any point, it can be seen that the error always increases in regions where $K \to \infty$ regardless of the numerical method being used.

The practical implementation of the geometric solution algorithm is described in detail in Appendix C.

4. CONCLUSION

In this paper we have presented a new method for solving Laplace's equation in exterior domains and in interior domains with complicated boundaries. This method will be regarded by many experts in numerical analysis as a strange method, since it is fundamentally different from all the previously known techniques. However, as is apparent from the analysis of Section 2 and from the experimental evidence presented, the method is especially suited for mapping the potential field throughout the entire domain of the problem and promises to be significantly faster than conventional methods which normally require an exhaustive amount of computations to achieve such a task. As was shown, the method requires an $O(N^2) + O(i \cdot M)$ computations, where N is the number of points used to obtain the initial Neumann conditions on the boundary, i is the number of iterations, and M is the number of points inside the domain at which the solution is computed. With i being a very small number, N being generally small, and with M being substantially large, the method essentially then requires an $O(M)$ computations and is therefore more advantageous than previous methods which require an $O(N \cdot M)$ computations, where both N and M are large.

A different and important implementation of the method would be to utilize the Rokhlin or other fast solvers to obtain the Neumann conditions at a large number of points N on the boundary. In such a case the relaxation algorithm described in Section 2.3 would not be necessary and the geometric method can be used as a tool for quickly mapping the field inside the domain of the problem. In such an implementation the total computational effort would be strictly $O(N) + O(M)$, which is still more advantageous than the $O(N \cdot M)$ effort required by the older techniques. Of course, the implementation is a matter of choice and is to be left to the preference of the user.

While fundamentally different from other presently known methods, we have shown that this method has its own merit in bringing large-scale calculations in complex domains within the limited computing power of desktop workstations and personal computers.

Finally, while the analysis of this paper presents the geometric solution of Laplace's equation in two dimensions, there is a straightforward extension to the method in three dimensions, which we will report at a later date.

APPENDIX A

LEMMA. *A closed equipotential surface cannot exist within a region where Laplace's equation holds.*

Proof. At every point in space where Laplace's equation holds, div $D = 0$, where $D = \varepsilon E$ is the electric flux density.

Consider a region where Laplace's equation holds. Consider a closed surface within that region. By applying the divergence theorem [11] to the volume of space enclosed by that surface, we have

$$\iiint_v \operatorname{div} D \cdot dv = 0 = \iint_S D \cdot dS, \tag{30}$$

where S denotes the area of the closed surface. But we have

$$E = -\text{grad}\, V$$
$$= -\frac{\partial V}{\partial n}\hat{n},$$

where $\hat{n}$ is a unit vector in the direction of the flux line. Therefore, assuming a uniform dielectric constant,

$$\iint_S D \cdot dS = -\varepsilon \iint_S \frac{\partial V}{\partial n}\hat{n} \cdot dS. \tag{31}$$

If, and only if, the closed surface is an equipotential surface, the electric flux will be normal to the surface everywhere, and $\hat{n} \cdot dS$ represents the dot product of a unit vector with itself times the area of the surface element dS; i.e., $\hat{n} \cdot dS = dS$ for every element on the surface.

Further, since equipotential surfaces cannot intersect, then, in the direction of the field lines, the potential V must be clearly decreasing outwardly everywhere on the surface; i.e., $\partial V/\partial n$ has a negative value for all elements on the surface (except, of course, the trivial case where V is constant; i.e., within a uniform conductor). Therefore, we conclude that the integral of the RHS of Eq. (31) cannot be equal to 0, and, hence, Eq. (31) contradicts Eq. (30).

APPENDIX B

Given an interval $x \in [a, b]$, a cubic spline function

$$f(x) = A_i + B_i(x - x_i) + C_i(x - x_i)^2 + D_i(x - x_i)^3 \tag{32}$$

defined over a sub-interval $x \in [x_i, x_{i+1}]$ (where x_i denotes the support points, $i = 0, ..., n + 1$) is completely determined in terms of its second derivatives (or *moments*) at the support points by [8]

$$\begin{aligned} A_i &= y_i \\ B_i &= \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{2M_i + M_{i+1}}{6}h_{i+1} \\ C_i &= \frac{M_i}{2} \\ D_i &= \frac{M_{i+1} - M_i}{6h_{i+1}}, \end{aligned} \tag{33}$$

where M_i is the moment at $x = x_i$ and $h_{i+1} = x_{i+1} - x_i$, $i = 0, ..., n$, is the partition size.

Further,

$$f'(x_i) = \frac{y_{i+1} - y_i}{h_{i+1}} - h_{i+1}\left(\frac{M_{i+1}}{6} + \frac{M_i}{3}\right). \tag{34}$$

Therefore, the curvature at the support points in an equipotential curve can be easily computed from

$$K_i = \frac{M_i}{(1 + [f'(x_i)]^2)^{3/2}}, \tag{35}$$

where the moments are first computed from the tri-diagonal system of equations [8]

$$\begin{pmatrix} 2 & \lambda_0 & & & & 0 \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & 2 & \lambda_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_n & 2 & \lambda_n \\ 0 & & & & \mu_{n+1} & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ \\ M_{n+1} \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \\ \alpha_{n+1} \end{pmatrix}, \tag{36}$$

where

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}$$
$$\mu_i = 1 - \lambda_i$$
$$\alpha_i = \frac{6}{h_i + h_{i+1}}\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right)$$

for $i = 1, ..., n$, with

$$\lambda_0 = 1, \quad \alpha_0 = \frac{6}{h_1}\left(\frac{y_1 - y_0}{h_1} - y'_0\right)$$
$$\mu_{n+1} = 1, \quad \alpha_{n+1} = \frac{6}{h_{n+1}}\left(y'_{n+1} - \frac{y_{n+1} - y_n}{h_{n+1}}\right).$$

For an equipotential curve, a difficulty arises at the two outermost points (the points located on the boundary), since the calculation of α_0, α_{n+1} requires the first derivative $y'(x)$, which is unknown at those points. However, a simplification can be made by observing a simple fact: if we assume that the second derivatives M_0, M_{n+1} at the end points of the curve are equal to zero, such assumption will not change the value of the first derivative at those points. In other words, taking $M_0 = M_{n+1} = 0$ simply assumes that the curve is "flat" at the end points.

By taking $M_0 = M_{n+1} = 0$, the tri-diagonal system is reduced to n equations. Such a system of equations can be solved in a straightforward fashion by Gaussian elimination [8], which results in a considerable saving in computation and storage (it is clear that only one array is needed to store the elements of the matrix, since $\mu_i = 1 - \lambda_i$).

APPENDIX C: Practical Implementation of the Geometric Relaxation Algorithm

In this section a description will be given for an algorithm that implements the geometric solution of Laplace's equation inside irregular domains. The algorithm is simple and is based directly on the basic equipotential construction method outlined in Section 2.1. The level of detail given here is such that an average physicist or engineer can clearly follow the description. Although the description may be lengthy, the algorithm is in fact quite compact and has been implemented in a few pages of C code. A pseudo-code listing of the equipotential tracing function of Section 2.1 will be given at the end of the Appendix; followed by a comment on the applicability of the algorithm to exterior domains.

Before describing the algorithm itself, an important notice about the algorithm is in order first. The implementation of this algorithm relies heavily on the technique of ''matching'' values of potential at different points on the boundary and inside the domain of the problem, that is, testing if two or more points have the same potential value. To implement this matching process correctly on real floating-point numbers, we shall define here a maximum error bound ε for the algorithm. This error bound is user-defined and represents the maximum error sought in the solution. For example, if the solution must be obtained with no less than 5-digit accuracy, then ε is set to be 10^{-5}; but if the solution is desired with only a 2-digit accuracy, then $\varepsilon = 10^{-2}$. The matching of the potential at different points, therefore, is intended to detect whether two or more points have the same value of potential within the specified error bound. This general rule will apply whenever statements like ''two points have the same potential,'' or ''V is the same'' appear in the following discussion.

The description given below will be aided by practical examples. The problems of Figs. 6 and 8 shown previously will be used as examples here.

C.1. Definition of Terminology

(a) *Start point.* A start point P_s is a point on the boundary at which the potential V is a local maximum or minimum; i.e., the potential either decreases monotonically or increases monotonically in either direction on the boundary around P_s (see Fig. C1a).

(b) *Start equipotential.* A start equipotential is a group of adjacent points on the boundary at which the potential V is unique and at which V is a local maximum or minimum (see Fig. C1b).

(c) *Inflection point.* It is the nearest point on the boundary to a start point or equipotential at which the sign of the normal potential gradient dV/dn is reversed (from positive to negative or vice versa, see Fig. C1c).

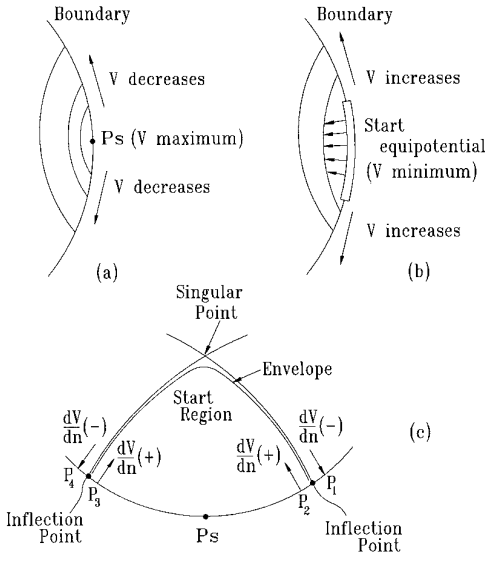

FIG. C1. Definition of various types of entities on the boundary or inside the domain of the problem: Start points, start equipotentials, inflection points, and start regions.

Each start point must be surrounded by two inflection points having exactly the same potential (except in the trivial case, where the entire boundary is an equipotential, as in Fig. 7). The potential of an inflection point always corresponds to and is equal to the potential of a singular point inside the domain, as shown in Fig. C1c.

(d) *Start region.* It is a region inside the domain of the problem surrounding each start point or equipotential. The last equipotential or ''envelope'' of that region is located at the inflection points. Such region is shown in Fig. C1c.

(e) *Intermediate region.* It is a region inside the domain of the problem which is not a start region. Each boundary value problem may contain zero or more intermediate regions. Intermediate regions exist when the domain of the problem contains more than one singular point. For illustration, the problem of Fig. 8 has eight start regions and two intermediate regions, while the problem of Fig. 6 has four start regions and no intermediate regions.

The definitions listed above will become more apparent in the following discussion.

C.2. Description of the Algorithm

The geometric relaxation algorithm consists of the following three major phases:

Phase A. Initialization of the Boundary Conditions

As explained in Section 2.3, the classical boundary-element method is typically used to obtain the initial Neumann conditions on the boundary. Let N be the highest

number of points which can be given on the boundary and at which the potential V can be defined. For low initial accuracy, solve for the Neumann conditions by using a subset of points N', $N' \subset N$, $N' \ll N$. For higher accuracy, N' must be higher, and the highest accuracy is eventually obtained when $N' \to N$. The boundary-element method is explained in detail in Ref. [17].

Phase B. Preprocessing the Boundary

This phase is concerned with locating the various entities on the boundary defined earlier, such as start points, inflection points, etc., and consists of the following five steps:

1. Considering the entire set of N points initially given on the boundary, scan the boundary to locate the start points and the start equipotentials. (Processing is simplified by giving each point on the boundary a unique ID number, from 1 to N, and by storing the coordinates of the boundary points and the values of potential at these points in three different arrays: X, Y, and V. The stored values for any point can then be accessed immediately by using the ID or sequential number of that point). As start points and equipotentials are being detected on the boundary, three arrays will be required: one array for recording the ID numbers of the start points found (for a start equipotential, only the first and last points on the equipotential will be recorded); the second array for indicating the type of each entity in the first array (i.e., whether the identified point is a stand-alone point or part of an equipotential); and the third array for indicating whether the entity is a local maximum or a local minimum. (Clearly, such task as identifying the type of a point can be simply accomplished by using two different digits or flags).

2. (a) For each of the boundary points 1 to N, excluding the subset of points N' identified in Phase A, calculate an estimated value for the normal potential gradient by linearly interpolating along the segment of the boundary containing the point, between the pair of values at the two points of the subset N' which surround that point. (By setting an array of N values for the normal gradient on the boundary, of which only a subset N' is known, this task is simply the task of filling in for the unknown values by using linear interpolation). (b) For each of the boundary points 1 to N, calculate the tangential potential gradient $\partial V/\partial x$, as $\partial V_i/\partial x = (V_{i+1} - V_{i-1})/\mathrm{Dist}_{i-1,i+1}$, where $\mathrm{Dist}_{i-1,i+1}$ is the physical distance between points $i-1$ and $i+1$ on the boundary. Calculate then the electric field intensity $\|E\|$ and the direction cosines δ_x, δ_y by using the set of formulas (3) in Section 2.1. (c) For each of the boundary points 1 to N, calculate the two quantities $\partial\|E\|/\partial x$, $\partial\delta_x/\partial x$, as the ratio of the difference over the physical distance between points $i-1$ and $i+1$, as explained above. Finally, calculate the quantity $\partial\|E\|/\partial s$ by using Eq. (11) in Section 2.1. (d) For each of the boundary points 1 to N, calculate the direction cosines of the electric field vector with respect to the global coordinates X, Y of the problem (note that we have used δ_x, δ_y to denote the direction cosines with respect to the local coordinates x, y on the boundary; see Fig. 2). Those direction cosines with respect to global coordinates will be denoted δX, δY and will be obtained from the following two formulas:

$$\begin{aligned}\delta X_i &= \delta_{x_i} * \frac{(X_{i+1} - X_{i-1})}{\mathrm{Dist}_{i-1,i+1}} - \delta_{y_i} * \frac{(Y_{i+1} - Y_{i-1})}{\mathrm{Dist}_{i-1,i+1}},\\ \delta Y_i &= \delta_{x_i} * \frac{(Y_{i+1} - Y_{i-1})}{\mathrm{Dist}_{i-1,i+1}} + \delta_{y_i} * \frac{(X_{i+1} - X_{i-1})}{\mathrm{Dist}_{i-1,i+1}}.\end{aligned} \tag{37}$$

The global direction cosines will be required later for constructing the equipotential curves progressively inside the domain as outlined in Section 2.1.

Clearly, this step will generate the following 11 arrays: $\partial V/\partial n$, $\partial V/\partial x$, $\|E\|$, δ_x, δ_y, $\partial\|E\|/\partial x$, $\partial\delta_x/\partial x$, $\partial\|E\|/\partial s$, δX, δY, and $\mathrm{Dist}_{i-1,i+1}$. (Note that the notation $\partial V/\partial n$ for the normal gradient will be used here instead of the notation $\partial V/\partial y$ used in Section 2.1).

3. Using the array $\partial V/\partial n$, scan the values of the normal potential gradient at the subset of points N' in order to estimate the location of the inflection points on the boundary. This can be performed in the following manner (see Fig. C1c for details): if the sign of the normal potential gradient (sgn(dV/dn)) at a particular point P_1 in the subset is negative, while sgn(dV/dn) at the following point P_2 is positive, or vice-versa, then an inflection point exist on the boundary at an undetermined location between points P_1 and P_2. Clearly, an inflection point is a point at which the sign of the normal potential gradient flips and which cannot be determined with high precision, since the resolution (N') used to obtain the Neumann conditions on the boundary is limited. The locations of the two points P_1 and P_2 on the boundary are then recorded. We shall hereinafter refer to such two points as the two ''characteristic'' points which indicate the presence of an inflection point on the boundary. A list of characteristic points is then created, such as P_1, P_2, P_3, P_4, etc., in the order of their respective locations on the boundary. Note that each start point or equipotential on the boundary must be surrounded by two inflection points, as shown in Fig. C1c, and as we indicated previously. The opposite, however, is not true; i.e., it is possible to have two or more inflection points occurring successively on the boundary with no start points or equipotentials lying therebetween (i.e., in Fig. C1c, it is not always necessary to find a start point P_s lying between two inflection points as shown). The reasons behind such rules of occurrence will be understood and clarified in the following discussion.

This step clearly requires only one array for recording the ID numbers of the characteristic points found.

Let us now see how steps 1–3 above will be applied to the examples of Figs. 6 and 8. In Fig. C2a, the preprocessing routine located four start points on the boundary: −1, +1, −1, +1. Each of the start points defines a start region inside the domain. Respectively, these regions are: *A*, *B*, *C*, and *D*. These regions are at this stage hypothetical regions, shown here only for clarity. The program also located eight characteristic points on the boundary, P_1 to P_8, each pair of characteristic points being around a corner point at which the potential is 0 V, as shown (clearly, these corner points are the precise locations of the inflection points). Note that in this particular example each start point found is surrounded by two inflection points and vice versa; that is, each inflection point is effectively "shared" between two different start points. The results obtained for the example of Fig. 8 are shown in Fig. C2b. As shown, the program located the following start points, anticlockwise from the bottom: +4.0 (equipotential), which defines region *A*; +1.0 (point), which defines region *B*; +2.0 (point), which defines region *C*; −1.0 (point), which defines region *D*; 0.0 (equipotential), which defines region *E*; −1.0 (point), which defines region *F*; +2.0 (point), which defines region *G*; and 0.0 (point), which defines region *H*. The characteristic points located by the program, P_1 to P_{24}, are also shown in the figure. In this example, some of the inflection points found are shared between start points or equipotentials, such as the inflection points defined by pairs P_3, P_4 and P_5, P_6; while some occur sequentially within a portion of the boundary that is free of start points or equipotentials, such as the sequence defined by pairs P_7, P_8; P_9, P_{10}; and P_{11}, P_{12}. We shall hereinafter refer to such points as the "free" inflection points (the occurrence of such points on the boundary indicate the presence of intermediate regions inside the domain).

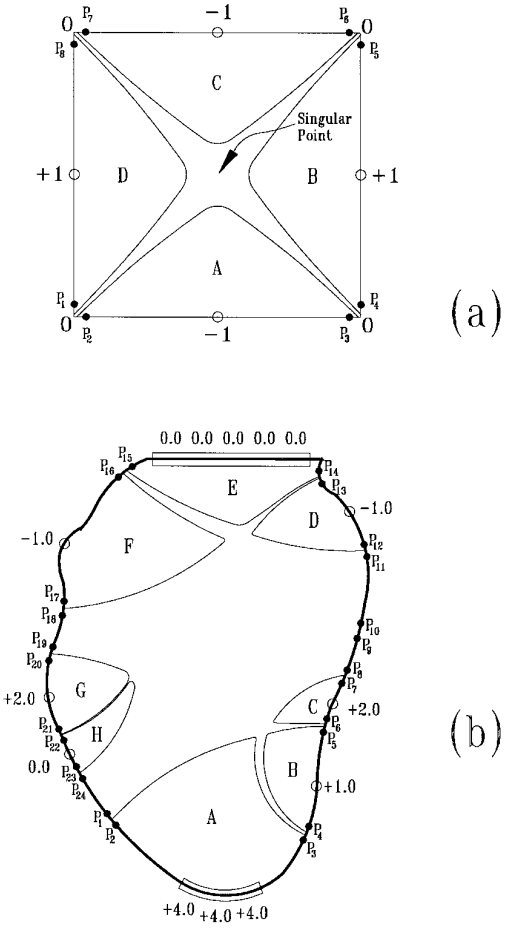

FIG. C2. The start points and the inflection points (indicated by the presence of pairs of characteristic points) found by the preprocessing routine for the examples of Figs. 6 and 8 shown previously. Each start point or equipotential defines a start region inside the domain.

4. The objective of this step is to determine those characteristic points which are shared between two different start points and finally form a "chain" of start points and equipotentials, such that each pair of characteristic points form a "link" between two different start points or equipotentials in the chain. In Fig. C2b, the pair of characteristic points P_3, P_4 located by the program is shared between start equipotential +4.0 and start point +1.0. The start point +1.0 is in turn linked to start point +2.0 by the pair P_5, P_6. On the other hand, pairs P_1, P_2 and P_7, P_8 are not shared between start points, as indicated above. It is therefore apparent that the sequence of start points +4.0, +1.0, and +2.0, and hence the corresponding regions *A*, *B*, and *C*, form one chain. Similarly, start points −1.0, 0, and −1.0 form another chain. Finally, start points +2.0 and 0.0 form a third chain (those chains may also be called clusters). The task of identifying those chains or clusters on the boundary is a straightforward task; by comparing the ID numbers of the start points with the ID numbers of the characteristic points (from the two arrays recorded previously), it is possible to determine those pairs of characteristic points which occur between start points or equipotentials (shared), and those which appear in sequence with no start points or equipotentials occurring therebetween (free). A group of start points or equipotentials which are found to be linked in this manner are then referred to as Group 1, Group 2, etc. To completely identify each group, three arrays will be required: two for the start points and equipotentials in that group (which must be arranged as explained in Step 1 above) and one for identifying the locations of the links, or the characteristic points, on the boundary. In more detail, the task is to be carried as follows: between each two consecutive start points P_{s1} and P_{s2} on the boundary, scan the list of characteristic points to determine if any pairs of characteristic points lie there between. If only one pair of such points is found, then a new group has been identified, and three new arrays

must be started as indicated above to identify the members of that group. If more than one pair, however, is found lying therebetween, then all the ID numbers of such characteristic points will be entered sequentially into one separate array listing the free inflection points on the boundary. If a following consecutive start point (or equipotential) P_{s3} is also found to be linked to P_{s2} as described above, that point or equipotential is added to the group, etc., until more than one pair of characteristic points is located, which indicates the end of that group.

For example, in Fig. C2b, the characteristic points found to be shared in the group *A-B-C* are P_3, P_4, P_5, P_6. The ID numbers of those points will then be entered into the proper array for that group. In Fig. C2a, all the pairs of characteristic points found are shared between start points as shown; hence, all the start points in this example form one group. It is very important to note here that a start point or equipotential which is not linked to any other start point or equipotential will form a group *by itself*, and therefore such a group will be composed of only one member.

5. The last step in the preprocessing routine is to calculate an inflection potential for each group of start points found in step 4. As can be easily seen in Fig. C2a, all four regions *A*, *B*, *C*, and *D* form one group which converges at a potential of 0 V, which is actually the inflection potential for each of the four regions separately. Now the purpose of ''grouping'' regions as explained in step 4 above is clear: each group of start points on the boundary which are linked in the manner described above must all have one, and precisely one, inflection potential (it is of course a simple matter to observe that if, for example, start points 1 and 2 share an inflection potential V_1 and start points 2 and 3 share an inflection potential V_2, while start point 2 must have only one value for the inflection potential, then V_1 must be equal to V_2). Therefore, each group of start points and equipotentials which are linked by inflection points as described above must all share one inflection potential; further, they must develop start regions which all terminate at a singular point inside the domain having a potential equal to that inflection potential. This fact is very apparent in the examples of Figs. 6 and 8. For illustration, the singular points inside the domain of Fig. 8 are shown in Fig. C3 below. As shown, there are three singular points in this example (compare with Fig. C2b and identify the corresponding start points and start regions for each singular point). As can be observed from the figure, each singular point is surrounded by a ''domain of influence,'' or a region where the potential is constant. These regions of constant potential are the black regions shown in the figure. The potential throughout each of those three regions is constant and is equal to the potential of the singular point at the middle to the region, which is also the potential of the corresponding inflection points on the boundary.

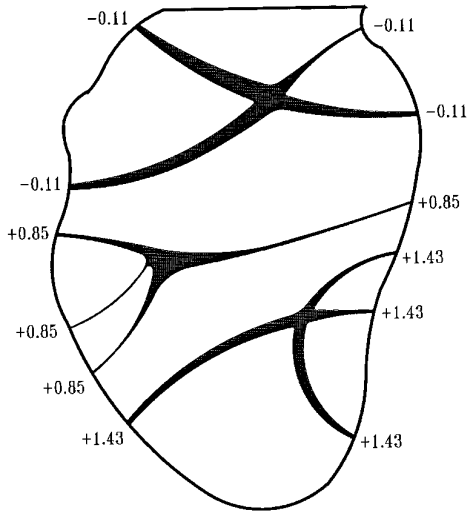

FIG. C3. The three singular points inside the domain of the problem of Fig. 8. Each singular point is surrounded by a ''domain of influence,'' or a region where the potential is constant (within an error bound of $\pm\varepsilon$) throughout such a region. As shown, each singular point has four inflection points on the boundary. The potentials of these singular points are: +1.43, +0.85, and −0.11 V.

When the solution converged inside the domain, these potentials were found to be: +1.43, +0.85, and −0.11 V, as shown in the figure.

To calculate a unique inflection potential V_{inf} for each group of start points found in step 4, we simply average the potentials of all the characteristic points listed for that group. For example, in Fig. C2b, the inflection potential for the group in the lower part of the domain (*A-B-C*) will be obtained by adding the values of the potentials at points P_1 through P_8, and dividing by eight. A unique V_{inf} will then be recorded for each group obtained. Note that such value of V_{inf} is only a first estimate and will be refined later by relaxation.

Steps 1–5 above describe the preprocessing routine. As can be observed, steps 1 and 2 will require an $O(N)$ work, where N is the number of points on the boundary. Steps 3, 4, and 5, combined, take a negligible amount of computations, since the tasks performed in those steps will require the handling of few arrays, composed of few points each. We can therefore conclude that this preprocessing procedure requires an $O(N)$ effort. Typically, the number of points N specified on the boundary is large (1000 or more); however, in comparison with the number of points M inside the domain at which the solution is to be computed, which is a quite substantial number (typically, tens of thousands of points), we see that the overhead of the preprocessing task is still negligible by comparison with the main task of solving the problem inside the domain (Phase C, described below). As we demonstrated in the example of Section

3.2.1, three iterations of the geometric relaxation algorithm took 25.3 s, which is now to be understood as an effort of $O(N) + O(3.M)$, where $N = 2000$ and $M = 62{,}000$. However, when the classical $O(N \cdot M)$ method was applied, the timing was 1.2 h. Clearly, the difference is substantial.

Phase C. Mapping of the Potential Field inside the Domain with Iterative Improvements of the Neumann Conditions

This is the main task which solves for the potential inside the domain, and is carried as follows.

1. *Tracing of equipotentials inside the start regions.* Starting at each start point or equipotential separately, implement the method described in Section 2.1 for tracing the equipotential curves progressively inside the domain. If the potential at a particular equipotential curve inside the domain is V_1, we must always select next two points on the boundary of a potential V_2, such that the step size $\Delta V = |V_1 - V_2|$ is much smaller than the minimum ratio $\|E\|/2K$ on the curve; as discussed in Section 2.2. The next equipotential curve will then be fitted through these two points on the boundary. When the solution starts at the start points and equipotentials, the curvature K at those points is assumed to be unknown, as indicated in Section 2.1. However, since the quantity $\partial\|E\|/\partial s$ has been calculated and since $\partial\|E\|/\partial s = -K\|E\|$ (Thomson's formula), then K also can be calculated; hence, the critical ratio $\|E\|/2K$ can be found at the start points. (It is important to observe that the curvature K may be a positive or a negative number, depending on the sign of the quantity $\partial\|E\|/\partial s$; however, only the magnitude of K is taken in the calculation of $\|E\|/2K$.) Note that when equipotential curves are traced progressively inside the domain, the potential will be decreasing if the start point is a local maximum and will be increasing if the start point is a local minimum. In both cases, the incremental distance Δs along the flux line will always be given by the ratio $\Delta V/\|E\|$. As the tracing of equipotentials progresses inside a start region, each support point on the most recently traced equipotential in that region will be given a unique ID number (from 1 to N) which corresponds to its origin or "root" on the boundary. All the support points obtained along the path or trajectory of a particular field line will then have a unique number, which is the number of the parent point on the boundary. For illustration, let a particular start point exist at location 253 on the boundary. The first equipotential traced inside the domain will then be composed of a total of three points (refer to Fig. 4). Let the boundary points selected for this equipotential be points 250 and 257. The point in the middle will then be given the number 253, which corresponds to the parent point on the boundary. The support points for that particular equipotential will then be numbered: 250, 253, and 257. This is a simple technique to monitor the origin of each support point obtained inside the domain.

Further details are given in the pseudo-code listing of the equipotential tracing function at the end of the appendix.

2. *Determining the "envelope" for each start region.* For each start region considered, the program monitors the value of the potential V on each new equipotential curve obtained inside the region. When it is determined that V has reached or passed the inflection potential V_{inf} for that region, the tracing of equipotentials is stopped in the region, and the last equipotential traced is considered to be the envelope for that region. For that purpose, if the start point or equipotential in the region is a local maximum, check if $V \leq V_{\text{inf}}$; otherwise check if $V \geq V_{\text{inf}}$. When such condition is met, the tracing in the region is stopped as indicated. The potential V of the envelope is then recorded. In addition, the coordinates of the support points of the envelope, together with the ID numbers of these points, are also recorded.

Figures C2a and b show the envelopes of the start regions for the two examples considered.

3. *Vestige elimination inside each group.* In this step preparation is made for: (a) tracing the intermediate regions inside the domain and (b) improving the Neumann conditions on the boundary at the origins of field lines, at those points on the boundary for which the trajectory of the field line has ended.

First, the presence of intermediate regions is detected by counting the number of groups found in step 4 of Phase B above. If the number of groups is 1, there are no intermediate regions inside the domain; otherwise intermediate regions exist. As can be observed, there are no intermediate regions in the problem of Fig. 6. In the problem of Fig. 8, two intermediate regions are present. Figure C4 below shows those two regions clearly (the intermediate regions are the two hatched regions shown inside the domain). The figure also shows the three distinct groups of start regions which were found in Phase B (numbered 1 to 3; compare with Fig. C2b). The task now is to form a new envelope for each group of start regions found, so that the tracing of the intermediate regions can start at the newly formed envelopes. A new group envelope is found by eliminating parts, or "vestiges," from the individual regions composing each group. For instance, consider Fig. C5, which shows an expanded view of the lower portion of Fig. C2b. As explained, the group A-B-C shown in the figure has the four characteristic points P_3, P_4 and P_5, P_6 shared between regions (as indicated previously, each pair of characteristic points may be called a link). Note that in the first few iterations the envelopes of neighboring regions may overlap, as shown in the figure. Now, for each envelope, select the point on the boundary which is closest to the link and form a vestige (or array of points) from that

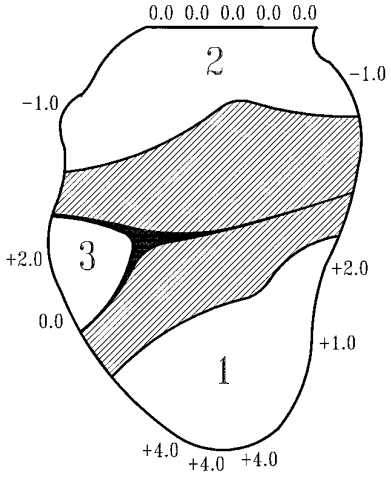

FIG. C4. The three distinct groups of start regions inside the domain of the problem of Fig. 8, with the envelope of each group shown as a thick curve. The two hatched regions are the intermediate regions inside the domain.

point and including all points on the envelope up to the support point which has the highest curvature K_{max}. For example, in Fig. C5, if the two boundary points of the envelope of region A are points 103 and 250, while the link points, that is, P_3 and P_4, are points 242 and 247, then boundary point 250 is selected. A vestige is then formed from that point up to and including the point of K_{max} shown on the envelope. This vestige may be called vestige AB. The portion of the envelope which was not selected (from the point next to K_{max} and up to point 103) is then entered into an array listing the support points of the new group envelope, as shown in the figure. In a similar manner, vestiges BA, BC, and CB are eliminated, and the remaining points (those shown on the envelope of region C) are entered into the final array representing the group envelope. This new group envelope is shown in heavy black on the outer boundaries of regions A and C. (Note that region B will be effectively annihilated in this process).

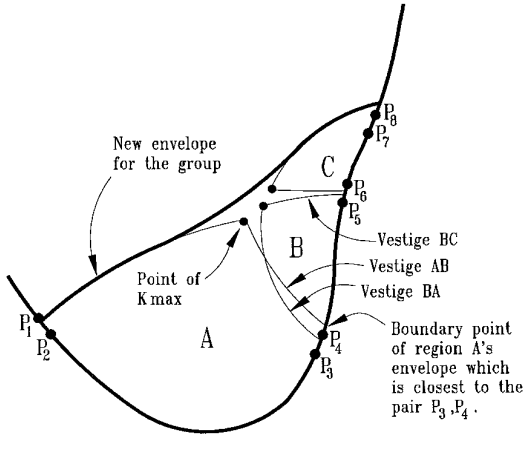

FIG. C5. Expanded view of the lower portion of Fig. C2b. The links, or pairs of characteristic points, P_3, P_4, and P_5, P_6 are shared among the regions A, B, and C. For each link, two vestiges must be eliminated. For the pair P_3, P_4, vestiges AB and BA are eliminated; and for the pair P_5, P_6, vestiges BC and CB are eliminated. In this process, the envelope of region B will be entirely annihilated, while the remaining portions of regions A and C's envelopes are entered into the array representing the new envelope of the group A-B-C.

This procedure must be repeated for every group of start regions inside the domain, and an array must be provided for each group envelope, listing the support points on that envelope (the three group envelopes which resulted from this process are shown clearly in Fig. C4). The eliminated vestiges will be each held in a separate array, in preparation for the next step. (In fact, a trio of arrays is required for each curve: X coordinates, Y coordinates, and ID numbers).

Note that when this procedure is applied to the example of Fig. C2a eight vestiges will be found (two for each start region) and no group envelope will be created (the array of the group envelope remains empty). It is of course possible, by the prior knowledge that no intermediate regions exist, to assert that no group envelope will be created.

4. *Potential averaging and feedback to improve the Neumann conditions.* In this step the potentials of the envelopes of neighboring regions will be averaged and used as a feedback factor in Eq. (22) of Section 2.3.2 to improve the estimation of the field intensity at the origins of the field lines on the boundary.

Before we explain the averaging process, we will recall that if the solution converges, all the start regions within one particular group must converge to one, and precisely one, inflection potential, which is yet unknown. Let us consider again the case of Fig. C2b above. We indicated in Phase B that the first estimation of V_{inf} for group A-B-C was given by the average of the individual potentials at points P_1 through P_8. We also indicated that, in the first few iterations, the envelope of each start region will be the equipotential that is obtained when V_{inf} is reached or passed (whether the potential is increasing or decreasing in that region). We must now point out that when the solution converges, and the correct V_{inf} is reached, that potential is *never* traversed by any envelope, but it is precisely the same (within an error bound of $\pm\varepsilon$), no matter how many equipotentials can be traced successively in each region. This will be immediately apparent by observing that the inflection potential is the potential of a singular point inside the domain, as shown in Fig. C3. The distance Δs between two successive equipotentials shrinks to zero in the neighborhood of a singular point (refer to Section 3.3 for an explanation), and, therefore, the singular point is never traversed by equipotentials. In Fig. C2a, each of

the four regions converge to precisely 0 V, which is the potential of the singular point at the center of the square.

Now, for the first iteration (refer to Fig. C5), the potential of A's envelope will be different from the potential of B's envelope, as we indicated previously. Since the two envelopes must converge to a unique value, the task now is to locate the parent points (on the boundary) of vestiges AB and BA and correct the field intensities at those points. Since each of the support points on vestiges AB and BA has a unique ID number which corresponds to its parent point on the boundary, then the locations of those parent points are immediately now at hand. The task now is this: for vestige AB calculate a normalized error *n.e.*, as *n.e.* $= (V_B - V_A)/V_B$. Similarly, for vestige BA, the normalized error will be $(V_A - V_B)/V_A$. The same applies to vestiges BC and CB. Once a normalized error term is calculated for each of these vestiges, that normalized error is used in Eq. (22) to improve the estimation of the field intensities at the parent points of the vestige (note that the error in potential, to a first order, is directly proportional to the error in field intensity). This process is illustrated more clearly in Fig. C6 below. In the figure, each support point is connected to its parent point by an arrow representing the trajectory of the field line inside the domain (the direction of the arrow only illustrates that the conditions at the support point are used as feedback to improve the conditions at the parent point). Since the normalized error term calculated for a particular vestige is the same for all the support points on the vestige, that normalized error term will also be the same for all the parent points of the vestige. Note that some of the parent points in regions A and C will be corrected, while *all* the parent points in region B will be corrected in this procedure. The remaining parent points in regions A and C clearly now belong to the new envelope of the group and will not be corrected at this stage. In Fig. C2a, all the parent points on the boundary will be corrected at this stage, since no intermediate regions exist in this example. As explained in Section 2.3.1, the direction cosines δ_x, δ_y of the electric field at the parent points must also be modified in addition to the intensity. Further, the global direction cosines δX, δY must be modified accordingly.

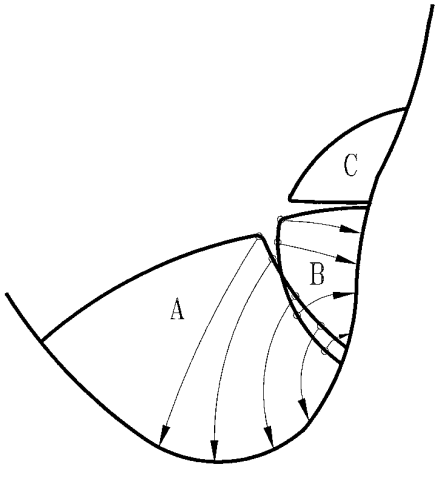

FIG. C6. The correction of the field intensities at the parent points of vestiges AB and BA shown in Fig. C5. Likewise, the parent points of vestiges BC and CB will be corrected. It is to be observed that only some of the parent points in regions A and C belong to the vestiges, while the remaining points in those two regions belong to the group envelope and will not be corrected at this stage. On the other hand, all the parent points in region B will be corrected in this procedure.

It is important to observe here that the geometric method is essentially a finite-difference method. The averaging and feedback technique used here is essentially the same as in other finite-difference methods. Only the global mechanism is different in this case.

5. *Calculating an improved inflection potential* V_{inf}. Following the improvement of the Neumann conditions described in the previous step, a new, improved value of V_{inf} must now be calculated for each group of start regions. To calculate this new V_{inf}, simply average the values of the potentials of all the individual envelopes in the group. In Fig. C6, the new V_{inf} for the group A-B-C will be the average of the potentials of the three envelopes shown in the figure.

6. *Tracing the intermediate regions.* This is the last step in the algorithm and consists of several small steps (if it was determined from step 3 that the problem has no intermediate regions, skip this step). As indicated previously, intermediate regions exist when the domain of the problem contains more than one singular point. In the simple case of Fig. 6, only one singular point exists inside the domain and, therefore, all the start regions converge at this singular point and no intermediate regions are created. In the more practical case of Fig. 8, three different singular points exist (shown clearly in Fig. C3). In order to trace the intermediate regions and, hence, cover the domain of the problem, proceed as follows:

1. Take the list of free inflection points obtained in step 4, phase B above. Organize the list such that the first point listed is the first point in Group 1 (if not already organized as such from Phase B). In the case of Fig. C2b, the list of free inflection points will be: $P_{1,2}$, $P_{7,8}$, $P_{9,10}$, $P_{11,12}$, $P_{17,18}$, $P_{19,20}$, and $P_{23,24}$ (note that the notation $P_{i,j}$ has been used as a shorthand for P_i, P_j, which designates the pairs of characteristic points). It is important to observe that the order in which groups are listed is arbitrary (the group A-B-C, or Group 1 shown in Fig. C4, can equally be labeled Group 2 or Group 3, depending on the order in which the program detects and labels the first group), and hence the inflection point $P_{1,2}$ could be associated with one of the other two groups shown in the figure.

2. For the list created in the previous sub-step, create a companion list showing the group number to which each of the free inflection points belong. In Fig. C2b, $P_{1,2}$ and $P_{7,8}$ belong to Group 1; $P_{11,12}$ and $P_{17,18}$ belong to Group 2; and finally $P_{19,20}$ and $P_{23,24}$ belong to Group 3. Note that the point $P_{9,10}$ does not belong to any group and will, therefore, be marked ''unassociated'' (with a special flag or digit).

3. Starting from the two points listed for Group 1 (the first two points in the list), search for the two nearest points on the boundary, which are also the two nearest points in the list. Figure C7a shows the list of free inflection points and shows that point $P_{9,10}$ is the nearest to point $P_{7,8}$. Since the boundary is closed, the point nearest to $P_{1,2}$ will be the one at the end of the list, which is $P_{23,24}$. Having determined the two nearest points, check the companion list to determine the group number with which each point is associated. As we shall observe, point $P_{9,10}$ is unassociated, while point $P_{23,24}$ belongs to Group 3. Such conditions will indicate that the list of free inflection points, as organized, is acceptable for processing. Another possible organization for the list is: $P_{11,12}$, $P_{17,18}$, $P_{19,20}$, $P_{23,24}$, $P_{1,2}$, $P_{7,8}$, $P_{9,10}$; in which Group 2 appears in the first place. Here, the two nearest points to Group 2 will be $P_{19,20}$, and $P_{9,10}$. Since $P_{19,20}$ is associated with Group 3, and $P_{9,10}$ is unassociated, such alternative organization is also acceptable for processing. A third possible alternative is: $P_{19,20}$, $P_{23,24}$, $P_{1,2}$, $P_{7,8}$, $P_{9,10}$, $P_{11,12}$, $P_{17,18}$; where Group 3 appears in the first place. In this case, however, the two nearest points are $P_{1,2}$, which belongs to Group 1, and $P_{17,18}$, which belongs to Group 2. Since the two nearest points belong to two different groups, such form of organization for the list will be unacceptable for processing. If such is the case, then the list of free inflection points must be reorganized such that Group 1 or Group 2 is on the top of the list. This can be simply achieved by moving the points of Group 3, $P_{19,20}$ and $P_{23,24}$, to the end of the list. The list will then appear as shown in Fig. C7a.

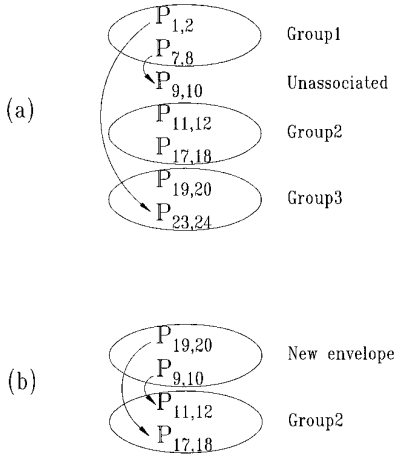

FIG. C7. The list of free inflection points on the boundary in the example of Fig. C2b. In the first sweep (a) to trace an intermediate region, the pair of characteristic points $P_{23,24}$ is used as a link, while the pair $P_{19,20}$ is not used. In the second sweep (b), $P_{19,20}$ is moved to the top of the list (see text), and finally both points in Group 2 are used as links; which signals the end of the task.

In summary, the organization of the list of free inflection points must be such that the two points which are nearest to the first group listed must be both associated with one group, or one is associated while the other is unassociated. The list then can be shuffled as explained above until this organization requirement is met.

4. Having organized the list as explained above and determined the two points which are nearest to the first group listed, start, from the envelope of the first group (in this case Group 1), tracing equipotentials progressively inside the domain (as explained in step 1, Phase C). For the case of Fig. C7a, we now observe the following: while the point $P_{9,10}$ is unassociated, $P_{23,24}$ belongs to Group 3, and therefore we conclude that the equipotentials progressing from Group 1 will terminate on the boundaries of Group 3 (these equipotentials will constitute the first intermediate region shown in Fig. C4). When the inflection potential V_{inf} listed for Group 3 is reached or passed, the tracing is stopped, and the last equipotential traced is recorded.

5. The two nearest points located in sub-step 3 above will now be used as links, which serve the purpose of vestige elimination. Since the point $P_{23,24}$ belongs to Group 3, and the point $P_{9,10}$ is unassociated, only $P_{23,24}$ will be taken as link in this case. Figure C8 shows the last equipotential traced in Intermediate Region 1 being located at or very close to the two inflection points $P_{23,24}$ and $P_{9,10}$. By taking the pair of characteristic points $P_{23,24}$ as a link, follow the procedure explained in step 3, Phase C, for eliminating two vestiges as shown in Fig. C8. As shown, one of the vestiges to be eliminated is a portion from the envelope of Group 3 and the other is a portion from the last equipotential traced in Intermediate Region 1. (Clearly, the vestige elimination procedure must preferably be a separate routine). For each vestige, calculate a normalized error term as explained in step 4, Phase C, and correct the Neumann conditions at the parent points of the vestige (again, this feedback procedure can be part of the separate routine). It can be observed from Fig. C2b that the parent points which will be corrected in this step are some of the parent points in regions H and A. It is very important to note that, as equipotentials are traced progressively inside the domain, the ID number of a given support point, which is the number of the parent point on the boundary, must be propagated to the next support point obtained along the field line (as explained in step 1, Phase C), so that the

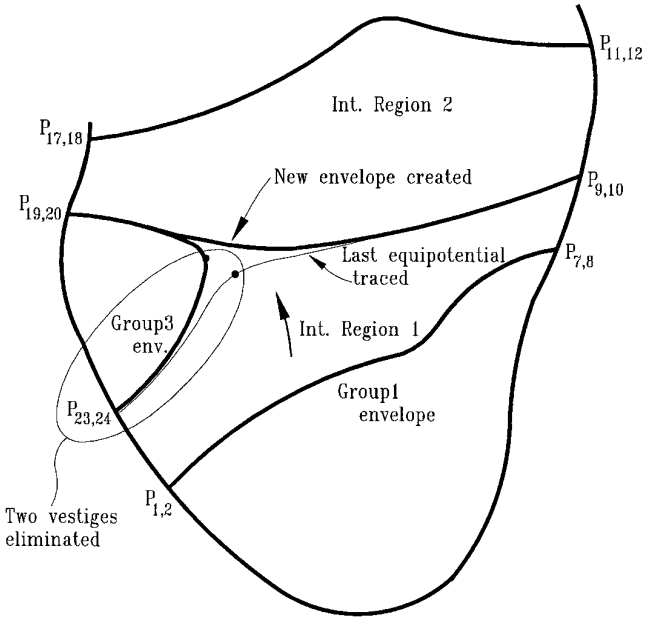

FIG. C8. The sweep to trace the first intermediate region inside the domain of Fig. C2b. The last equipotential traced in that region is obtained when V_{inf} of the target group (Group 3) is reached. Two vestiges are then eliminated at the location of the link $P_{23,24}$, and a new envelope is created. From that envelope, a second sweep is initiated to trace Intermediate Region 2.

location of the parent point can be immediately identified at any stage in the procedure. The vestige elimination routine will finally deliver the new envelope shown in Fig. C8.

6. We now make reference to Fig. C7b. The objective now is to eliminate from the list of free inflection points the two points $P_{1,2}$ and $P_{7,8}$ which belong to Group 1 and, also, any points which have been used as links (in this case, $P_{23,24}$). Note that point $P_{9,10}$ was not used as link and, therefore, will not be eliminated from the list. Next, the remaining point in the target group (Group 3) which was not used, point $P_{19,20}$, is moved to the top of the list. The modified list is shown in Fig. C7b. (Had both points in the target group been used as links, this would indicate the end of the task and the end of the algorithm, as will be explained shortly). The two points on the top of the list now belong to the new envelope, as can be seen in Fig. C8. The same elimination and rearranging procedure must also be performed on the companion list that shows the group with which each point is associated (the two points belonging to the new envelope must be marked with a special flag or digit).

7. Starting from the newly obtained envelope, repeat sub-steps 4 to 6 above, until *both* free points in the target group are used as links. In Fig. C7b, the target group is Group 2, and, as shown, point $P_{17,18}$, being at the end of the list, is selected. However, since point $P_{11,12}$ of that group follows immediately point $P_{9,10}$, that point is also selected; hence, both free points in Group 2 will be used as links in sub-step 5 above. Sub-step 6 will then eliminate all entries in the list of free inflection points. Note that each sweep through sub-steps 4 to 6 traces one intermediate region. Therefore, in Fig. C8, two sweeps will be performed in order to trace the two intermediate regions shown (which can also be concluded from Fig. C7).

Iterations. Repeat Phase C (steps 1 through 6), until a prespecified error criteria is met (the error between two successive iterations is smaller than a prespecified increment. This prespecified increment must preferably be equal to or smaller than the error bound ε used throughout the algorithm).

C.3. Pseudo-Code of the Equipotential Tracing Function

The basic equipotential tracing function described in Section 2.1 and used in steps 1 and 6, Phase C, is listed here in detailed, pseudo-code form:

(a) For a given start point P_s or a start equipotential, calculate the minimum critical ratio $\|E\|/2K$, as described in step 1, Phase C, and select a step size of potential ΔV by dividing that critical ratio by some constant (preferably 10 or higher; note that such a constant controls the accuracy of the solution). Select then two points on the boundary around P_s of a potential V_2 such that $|V_1 - V_2| \leq \Delta V$, as described in step 1.

(b) Calculate a displacement $\Delta s = \Delta V/\|E\|$ at the start point, as described in Section 2.1; or, for a start equipotential, calculate several displacements, one for each point defined on the equipotential. Calculate then the X and Y coordinates of the support points on the new equipotential inside the domain, as well as the ID numbers at these points, by using the following three formulas:

$$\begin{aligned} X(\text{new})[k+1] &= X(\text{old})[k] + \Delta s[k] * \delta X[k], \\ Y(\text{new})[k+1] &= Y(\text{old})[k] + \Delta s[k] * \delta Y[k], \\ \text{ID}[k+1] &= \text{ID}[k], \end{aligned} \tag{38}$$

where k is a counter (from 0 to $n - 1$) and δX, δY are the global direction cosines calculated in step 2, Phase B. Note that the order of points on the new equipotential has been shifted by one to allow for the data (X, Y, and ID) of the two newly selected boundary points to be inserted at locations [0] and [$n + 1$] in the arrays.

(c) For the newly obtained support points inside the domain, calculate the field intensity at each of these points by using Eq. (9) of Section 2.1, expressed as

$$\|E\|(\text{new})[k+1] = \|E\|(\text{old})[k] * (1 - \Delta s[k] * K[k]), \quad (39)$$

$k = 0$ to $n-1$, where K is the curvature calculated and used in (a) above. (Note that the sign of K always determines whether $\|E\|$ will be decreasing or increasing.) At the two boundary points 0 and $n+1$ of the new equipotential, the field intensity $\|E\|$ was calculated in step 2, Phase B, and is already available at those points.

(d) For the newly obtained support points inside the domain, 1 to n, calculate the values of λ and α (see Appendix B) from the following two formulas:

$$\lambda[k] = \frac{X[k+1] - X[k]}{X[k+1] - X[k-1]},$$
$$\alpha[k] = \frac{6}{X[k+1] - X[k-1]} * \left(\frac{Y[k+1] - Y[k]}{X[k+1] - X[k]} - \frac{Y[k] - Y[k-1]}{X[k] - X[k-1]}\right). \quad (40)$$

Then, call a Gaussian elimination routine to obtain the values of the moments at the n points, $M[1]$ to $M[n]$. Finally, set $M[0] = M[n+1] = 0$, as described in Appendix B.

(e) By using the array of moments $M[0]$ to $M[n+1]$ obtained in (d), calculate the first derivative of the function $y = f(x)$ at all the support points 0 to $n+1$ by using formula (34) of Appendix B, expressed as

$$y'[k] = \frac{Y[k+1] - Y[k]}{X[k+1] - X[k]} - (X[k+1] - X[k]) * \left(\frac{M_{k+1}}{6} + \frac{M_k}{3}\right), \quad \text{for } k = 0 \text{ to } n;$$
$$y'[k] = \frac{Y[k-1] - Y[k]}{X[k-1] - X[k]} - (X[k-1] - X[k]) * \left(\frac{M_{k-1}}{6} + \frac{M_k}{3}\right), \quad \text{for } k = n+1. \quad (41)$$

Now, calculate the curvature K at the support points 1 to n of the new equipotential by using formula (35) of Appendix B. For the two boundary points 0 and $n+1$, calculate K by using the quantity $\partial\|E\|/\partial s$ known at those points, as $K = -(\partial\|E\|/\partial s)/\|E\|$.

(f) The final calculation in this procedure is to obtain new direction cosines which represent the directions of the electric field vectors at the newly obtained support points inside the domain. For that purpose, two new arrays of $n+2$ values each will be required. Let such two arrays be named $\delta X_{\text{internal}}$ and $\delta Y_{\text{internal}}$. If the new equipotential has directly resulted from a start point or equipotential on the boundary, initialize the two arrays by copying the values of δX, δY on the boundary into the new arrays. The locations 0 and $n+1$ in those arrays will be filled by the values of δX, δY already available at the two newly located boundary points. Those arrays will be now destroyed and refilled with new values for the new equipotential; and this process will be repeated for each new consecutive equipotential curve obtained inside the domain. The following lines of code will calculate the correct direction cosines of the electric field vectors at the new support points 1 to n, given that the old values are present in the arrays:

```
for(k = 1 to n)
    θ = arc tan(y'[k]) + 1.57079632;
    if(cos θ * δX_internal + sin θ * δY_internal) < 0
      let θ = arc tan(y'[k]) − 1.57079632;
    δX_internal = cos θ;
    δY_internal = sin θ;
```

The **if** statement in the above procedure checks if the angle between the old and the new vectors is larger than 90° (an impossibility), and adjusts the new angle accordingly, using the slope $y'(x)$ of the tangent to the equipotential. The procedure then replaces the old values present in the arrays with the new values calculated.

(g) Repeat procedures (a) to (f) described above to trace each new equipotential curve inside the domain.

C.4. Applicability of the Algorithm to Exterior Domains

Problems in exterior domains, in general, are not as trivial as the simple example shown in Fig. 7. The classical approach for performing computations in exterior domains used in various methods is to divide the open domain of the problem into two or more closed subdomains and to solve inside each of these subdomains as in a standard interior problem. Figure C9 below shows an irregular boundary, which may be the surface of an airfoil, for example. As shown, the open domain around the boundary is

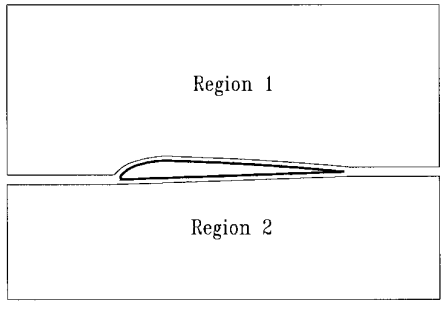

FIG. C9. The subdivision of an exterior domain into two interior domains with closed boundaries.

subdivided into two separate, closed regions, where the conditions along the boundary of each region are known, as, for example, the steady-state conditions at a large distance away from the airfoil. Each of the two regions can then be treated as a separate interior problem.

Such a well-known approach for solving boundary value problems in exterior domains can be equally used in the case of the geometric method without any difficulty.

ACKNOWLEDGMENTS

We thank Professors H. Greenside, R. Bryant, P. Marinos, D. Overhauser, and D. McCumber for many useful discussions. We also thank a referee for reviewing this work in detail. Special thanks to Dr. Leslie Greengard for reviewing the paper and for providing some references and other valuable information.

REFERENCES

1. E. G. Bakhoum and J. A. Board, Jr., *Phys. Rev. E* **47**(5), 3682 (1993).
2. E. G. Bakhoum, Ph.D. dissertation, Duke University, 1994.
3. V. Rokhlin, *J. Comput. Phys.* **60**(2), 187 (1985).
4. A. Greenbaum, L. Greengard, and G. B. McFadden, *J. Comput. Phys.* **105**(2), 267 (1993).
5. A. Mayo, *SIAM J. Sci. Stat. Comput.* **6,** 144 (1985).
6. T. Palmer, E. Simpson, K. Kavanagh, and W. Smith, "Visualization of Bioelectric Phenomena," in *High Performance Computing in Biomedical Research,* edited by T. C. Pilkington (CRC Press, Boca Raton, FL, 1993).
7. M. P. DoCarmo, *Differential Geometry of Curves and Surfaces* (Prentice–Hall, Englewood Cliffs, NJ, 1976), p. 16.
8. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer-Verlag, New York, 1980), p. 93.
9. R. Varga, *Matrix Iterative Analysis* (Prentice–Hall, Englewood Cliffs, NJ, 1962).
10. D. Young, *Iterative Solution of Large Linear Systems* (Academic Press, New York, 1971).
11. W. H. Hayt, Jr., *Engineering Electromagnetics* (McGraw–Hill, New York, 1981).
12. J. Clerk Maxwell, *A Treatise on Electricity and Magnetism* (Dover, New York, 1954), p. 154.
13. T. N. E. Greville, "Introduction to Spline Functions," in: *Theory and Applications of Spline Functions,* edited by T. N. E. Greville (Academic Press, New York, 1969).
14. J. D. Jackson, *Classical Electrodynamics* (Wiley, New York, 1975), p. 51.
15. G. A. Estevez and L. B. Bhuiyan, *Am. J. Phys.* **53,** 133 (1985).
16. R. Pappas, *SIAM Rev.* **28,** 225 (1986).
17. C. A. Brebbia, *The Boundary Element Method For Engineers* (Pentech Press, London, 1984).
18. Research and Education Association, *The Numerical Analysis Problem Solver* (REA, New York, 1986).
19. A. Brandt, *Math. Comput.* **31,** 333 (1977).
20. W. Hackbusch, *Multigrid Methods and Applications* (Springer-Verlag, New York, 1985).
21. W. H. Press *et al.*, *Numerical Recipes in C* (Cambridge Univ. Press, New York, 1988).
22. W. Tang, *Transforming Domain into Boundary Integrals in BEM* (Springer-Verlag, New York, 1988).
23. M. N. Sadiku, "A Comparison of Numerical Methods for Computing Electromagnetic Fields," in *Proceedings, Southeastcon., 1990.*