An Adaptive Grid Scheme Using the Boundary Element Method

RAMAKANTH MUNIPALLI AND DALE A. ANDERSON

Department of Mechanical and Aerospace Engineering, University of Texas at Arlington, Arlington, Texas 76019

Received April 17, 1995; revised April 15, 1996

A technique to solve the Poisson grid generation equations by Green's function related methods has been proposed, with the source terms being purely position dependent. The use of distributed singularities in the flow domain coupled with the boundary element method (BEM) formulation is presented in this paper as a natural extension of the Green's function method. This scheme greatly simplifies the adaption process. The BEM reduces the dimensionality of the given problem by one. Internal grid-point placement can be achieved for a given boundary distribution by adding continuous and discrete source terms in the BEM formulation. A distribution of vortex doublets is suggested as a means of controlling grid-point placement and grid-line orientation. Examples for sample adaption problems are presented and discussed. © 1996 Academic Press, Inc.

INTRODUCTION

Grids that adapt to the developing features in flow-fields during numerical calculations represent the physics of the problem being studied. For problems where unstructured grids are used, adaption can be accomplished relatively inexpensively because insertion of additional grid points is the most popular way to achieve refinement. For structured grids, both grid-point motion and refinement schemes are used. When motion methods are used, the adaption may consume as much as one-third of the total time spent doing the simulation. For this reason, it is important to develop new methods of grid adaption that may be used more efficiently.

Structured meshes can be obtained using a variety of algebraic or partial differential equation methods. Winslow [1] used Laplace's equation to generate smooth grids used in solving the heat equation. Thompson *et al.* [2] added source terms to the right-hand side with the goal of proving control of mesh point location and density in the physical domain. Barfield [3], Amsden and Hirt [4], and Godunov and Prokopov [5] also solved Poisson related equations to obtain grids used in digital simulations. Limited success was attained in achieving desired grid point clustering and this led to interesting and highly unique research in solving this problem. Thomas and Middlecoff [6] developed a method of controlling interior clustering based on boundary point distributions. Rai and Anderson [7] presented a scheme based on an attraction model between points based on analogy with gravitational fields. Brackbill and Saltzman [8] solved the variational equations that optimize a prespecified quantitative measure of the required grid properties to produce adaptive grids. Anderson [9, 10] first developed the analytical relationship between the source terms in the Poisson grid generators and grid point clustering, opening the way for true adaptive control using elliptic grid generators. Liao and Anderson [11] recently applied concepts from deformation theory, originally proposed by Moser [12], to generate grids including a guarantee of uniqueness and existence.

The conventional approach of generating grids using Poisson equations is to transform the equations to logical space, resulting in a system where the role of the dependent and the independent variables has been reversed. With this idea, the physical coordinates are directly computed and the numerical computation is simplified since the spacing in computational space ($\Delta\xi$, $\Delta\eta$) may be set equal to a constant. The discrete form of the transformed Poisson equation is greatly simplified by this procedure. The disadvantage of using the transformation to logical space is that the Poisson equation, linear in physical space, is nonlinear in computational space. Furthermore, no guarantee of uniqueness or existence can be made in general for this case.

The problems that result from the necessity of computing a solution of a nonlinear partial differential equation suggest that other approaches to computing solutions to the Poisson equation be explored. In an earlier paper [13], the Poisson equation was solved in the physical domain, where it remains linear, and the physical coordinates were determined by interpolating for the constant computational coordinate lines. In the same work, the solution of the Poisson equation was obtained using Green's functions and a further simplification was made by employing approximations to the Green's functions in two and three dimensions. In this paper, the use of Green's functions is extended by introducing the boundary element formulation in the solution method and significant reductions in the order of the problem are obtained.

0021-9991/96 $18.00

LINEARIZED ADAPTION

The standard Poisson elliptic grid generation equations written with physical coordinates as the independent variables are of the form

$$\begin{aligned}\nabla^2\xi &= P\\ \nabla^2\eta &= Q.\end{aligned} \tag{1}$$

These equations may be integrated over any arbitrary volume to yield the integral or finite-volume form

$$\int_{\mathrm{vol}} \nabla^2\xi\, d(\mathrm{vol}) = \oint_{\partial\,\mathrm{vol}} \nabla\xi\cdot d\bar{s} = \int_{\mathrm{vol}} P d(\mathrm{vol}). \tag{2}$$

This formulation shows that the source functions, (P, Q), act as sources for the creation of ξ and η and thus control the grid spacing.

In the conventional Thompson scheme, the Poisson equation is written with the roles of the independent and the dependent variables reversed. When this transformation is completed, the equations take the form

$$\alpha\bar{r}_{\xi\xi} - 2\beta\bar{r}_{\xi\eta} + \gamma\bar{r}_{\eta\eta} = -J^2(P\bar{r}_\xi + Q\bar{r}_\eta), \tag{3}$$

where the quantities α, β, γ, and J are given by

$$\begin{aligned}&\alpha = x_\eta^2 + y_\eta^2, \quad \beta = x_\xi x_\eta + y_\xi y_\eta, \quad \gamma = x_\xi^2 + y_\xi^2,\\ &J = x_\xi y_\eta - x_\eta y_\xi\end{aligned} \tag{4}$$

and

$$\bar{r} = (x, y)^{\mathrm{T}}$$

The transformation of the original linear equation (Eq. (1)) from physical space to computational space produces a highly nonlinear system of equations that may be directly solved for the physical coordinates. In order to retain linearity in this system, the original system must be retained with the source terms (P, Q) written as functions of (x, y). With (P, Q) written as functions of (x, y), the original system is homogeneous and decoupled. We propose to exploit these properties in creating an efficient solution technique. The homogeneity of the system permits the Green's functions to be very simply obtained. With this approach, the final form of the governing grid generation equations for a two-dimensional problem may be written

$$\frac{\partial^2\xi}{\partial x^2} + \frac{\partial^2\xi}{\partial y^2} = P(x, y), \frac{\partial^2\eta}{\partial x^2} + \frac{\partial^2\eta}{\partial y^2} = Q(x, y) \tag{5}$$

for the domain, D, and on the boundary the values of ξ and η are prescribed as

$$\xi = \xi_0(x, y), \quad \eta = \eta_0(x, y). \tag{6}$$

The choice of P and Q is crucial because these source functions determine the control of mesh point spacing. Sophisticated methods of specifying the source functions have been developed when the Thompson scheme is used, permitting the direct computation of the physical coordinates from Eq. (3). Unfortunately, these formulations cannot be used for the linear problem described here. However, it is instructive to approximate these source functions as closely as possible for the linear case. As an example, the source terms may be approximated (for an x, y aligned system) as

$$P = Q = \frac{1}{w}|\nabla w|. \tag{7}$$

If the coordinate system is not aligned with the (x, y) system, a more logical definition in general may be written

$$P = \frac{1}{w}\nabla w\cdot\nabla u \tag{8}$$

with a corresponding definition for Q, where the weight function, w, may be written

$$w = 1 + A|\nabla u|^2 \tag{9}$$

and u is some physical quantity like pressure, density, or an appropriate combination that may be used to cluster points.

This system is solved to obtain the values of ξ and η at the cell centers. Grid point locations correspond to the integer values of ξ and η and interpolation is necessary to obtain the new grid point locations. This process is initiated within the current grid by starting with a grid point whose exact location in the final grid is known (usually the lower left-hand corner point). Layers of cells surrounding this point are scanned for the grid point that is sequentially next in order. The search path is depicted in Fig. 1. Since the ultimate goal in using the linearized form of the mesh generation equations is to reduce the CPU time, it is appropriate to consider the time required to perform the search for target cells. Using the above idea, a grid point is found in a row in the neighborhood of the initial point. A number, α, exists such that the next grid point can be found after visting less than α cells. This implies that the operation count for this search through the entire domain will be less than $\alpha\mathbf{N}$, where $\mathbf{N}$ is the total number of grid points in the domain. This is an $O(N)$ process.

When a host cell has been located that contains the target point, bilinear interpolation is carried out to fix the point location within this cell. This procedure fixes the (x, y) values for the point in the computational domain. Bilinear interpolation requires the solution of a quadratic

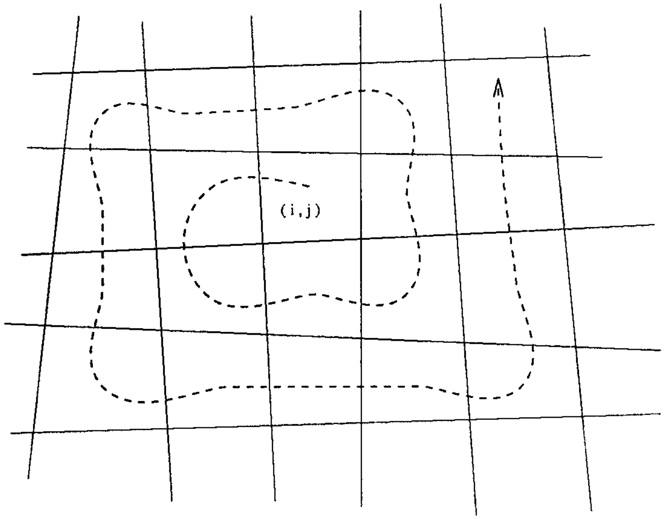

FIG. 1. Search path for grid point location.

equation but produces second-order accuracy. The procedure detailed here is fast, robust, and produces smooth grids that adapt well to the features of the evolving solutions. Linear interpolation was also attempted but the final grids were not as smooth, nor did they adapt as accurately to the flow quantities defining the desired mesh point distributions.

Conceptually, the idea presented above can easily be extended to three dimensions. In that case, a third equation is added to the grid generation equations, (Eq. (1)), resulting in the system

$$\begin{aligned}\nabla^2\xi &= P\\ \nabla^2\eta &= Q\\ \nabla^2\zeta &= R.\end{aligned} \tag{10}$$

After this system of equations is solved for the computational coordinates, the search for the host cell proceeds in the same manner as in the two-dimensional case. However, the interpolation routine requires modification and the process of locating a point in a cell is significantly more difficult. In three dimensions, the volumes of the tetrahedra formed by joining the target point and the test cell vertices is formed. If the sum of these volumes is equal to the volume of the test cell, the point lies within the cell. The values of the (x, y, z) coordinates of the point are established by linear interpolation in (ξ, η, ζ) space. The extension of bilinear interpolation into three dimensions produces a very complex system and for the purposes of this paper was not pursued.

Solution Using Green's Functions

The solution of boundary value problems using Green's functions is a classical technique (Garabedian [14]). By way of review, consider the homogeneous, linear PDE,

$$\mathbf{L}u = h(x, y), \tag{11}$$

subject to appropriate boundary conditions. Define the Dirac delta function as

$$\delta(x - x_0, y - y_0) = \begin{cases} 1 & \text{if } (x, y) = (x_0, y_0)\\ 0 & \text{otherwise.}\end{cases} \tag{12}$$

The delta function represents a unit impulse acting at a given point in the domain. The Green's function is defined as the solution of the PDE,

$$\mathbf{L}G = \delta(x - x_0, y - y_0). \tag{13}$$

With this notation, the solution to Eq. (11) is given by

$$u(x, y) = \int\int_S h(x, y)G(x, y : x_0, y_0)\, dx_0\, dy_0. \tag{14}$$

The use of Green's functions requires that a solution be computed to the system of equations,

$$\begin{aligned}\nabla^2\xi_G(x_0, y_0 : x, y) &= \delta(x - x_0, y - y_0)\\ \nabla^2\eta_G(x_0, y_0 : x, y) &= \delta(x - x_0, y - y_0),\end{aligned} \tag{15}$$

where ξ_G and η_G are the Green's functions at (x, y) for a unit impulse at (x_0, y_0). The discrete system that must be solved for the Green's functions at each point in the domain may be written

$$\begin{aligned}\{\mathbf{L}\}|\xi_G(x_0, y_0 : x, y)] &= \delta(x - x_0, y - y_0)\\ \{\mathbf{L}\}[\eta_G(x_0, y_0 : x, y)] &= \delta(x - x_0, y - y_0).\end{aligned} \tag{16}$$

In this expression, the left side represents an $m \times n$ matrix, where m and n are the number of grid points in the x and y directions respectively.

An alternative to computing Green's functions using these methods is to construct analytic approximations to the Green's functions. These approximations include $\log(r)$ and $1/r$ for two and three dimensions, respectively, and give close estimates when properly scaled. This approach is suggested by the solution techniques applied to irrotational, inviscid, incompressible flow where sources, vortices etc. are superimposed on a free-stream flow. For example, a point vortex is a unit impulse applied for Green's function computation. In two-dimensional problems, Green's functions, ξ_G, are written as the sum of two components, ξ_g and ξ_L, given by

$$\nabla^2 \xi_L = 0 \qquad \text{for } r \geq 0 \tag{17}$$

$$\xi_g = \log\left(\frac{r}{r_o}\right) \qquad \text{for } r \leq 0, \tag{18}$$

where the solution for ξ_G is written

$$\xi_G = \xi_L + \xi_g \qquad \text{for } r \leq r_0 \tag{19}$$

$$\xi_G = \xi_L \qquad \text{for } r \geq r_0. \tag{20}$$

The quantity r_0 represents the range of influence of the unit impulse and this range may be problem dependent. In general, this is the straight line distance between any two points in the domain. This forces the Green's function to vanish far away from the location of the impulse. In this method, the influence functions are directly added to the local value of ξ. It is then possible to easily avoid grid lines overlapping each other by making sure that (to some level of tolerance):

$$(1 + \xi_G(i, j)) < (\xi_G(i + 1, j)).$$

The Poisson equation is elliptic and no preferred direction for signal propagation exists. Thus, the distance, r, is the distance between two points in a smooth, simply-connected domain that limits the influence of an applied impulse. For a problem covering the entire n-dimensional space ($n > 2$), the Green's function is given by

$$\xi_{\text{Green}} = \frac{1}{\sigma'_{n-1}} \frac{1}{r^{n-2}}, \tag{21}$$

where $\sigma'_{n-1} = (n - 2)\sigma_{n-1}$, with σ_{n-1} the surface area of the $(n - 1)$-dimensional unit sphere [14]. For the case of an infinite domain, the Euclidean metric,

$$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{22}$$

is the required distance. The presence of boundaries necessitate changes in this representation of the Green's function. Consider the solution to the Laplace equation in the upper half plane, $z \geq 0$. The Green's function in this case, is given by

$$G = \frac{1}{\sigma'_{n-1}} \left[\frac{1}{r^{n-2}} - \frac{1}{\tilde{r}^{n-2}}\right], \tag{23}$$

where r is the distance from the point, (x, y), to the point, (x_0, y_0), and $\tilde{r}$ is the distance from the reflection of (x_0, y_0) to (x, y). Garabedian [14] shows the extension of these Green's functions to several simple domains that are related by bilinear mappings through the method of reflection. This is identical to the method of images used in aerodynamics.

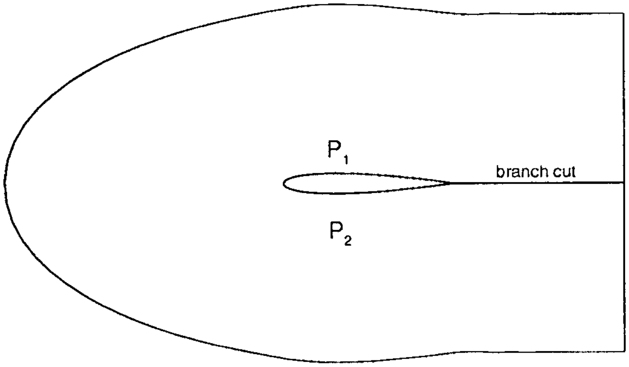

FIG. 2. Domain for C-grid around an airfoil.

The application of this approach poses some challenging problems because the method of images is not easily applied to complex domains. Even for a case as simple as a triangle, the application is cumbersome due to the complicated form of the analytic Green's function. In view of this, a simplified Green's function approximation that contains the essential features of the fundamental solution justifies investigation. Consider the form

$$\xi_G = \xi_L + \kappa \log\left(\frac{r}{r_0}\right), \tag{24}$$

where the boundary conditions are satisfied by this function; i.e., the effect of the unit impulse dies down at the boundary. The boundary conditions can be accounted for by initially adjusting the boundary points to suit the adaption requirements and basing the remainder of the adaption on boundary point final location. This eliminates the influence of the interior points on the movement of the boundary points. Concave domains present a more difficult problem. Figure 2 shows a C grid around an airfoil. Point P_1 on the upper surface of the airfoil influences the point P_2 on the lower surface of the airfoil. However, the distance that normally is used in producing the decay of this influence is no longer the Euclidean metric as used in the previous discussion. Instead, the effect of the curvature of the surfaces must be included in the distance functions.

The effect of having curved boundaries precludes the use of the Euclidean metric, except in the case of a branch cut where smoothness of the grid lines is desired. Some measure, r, that best describes the influence of points and the associated decay of this influence is needed. In the

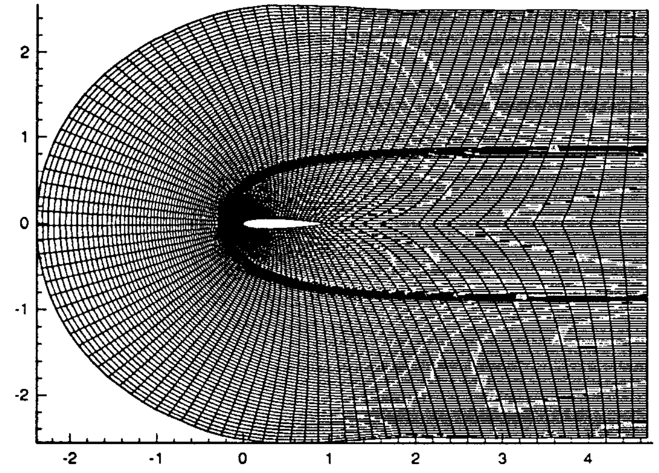

FIG. 3. Grid adapted to an η-like line.

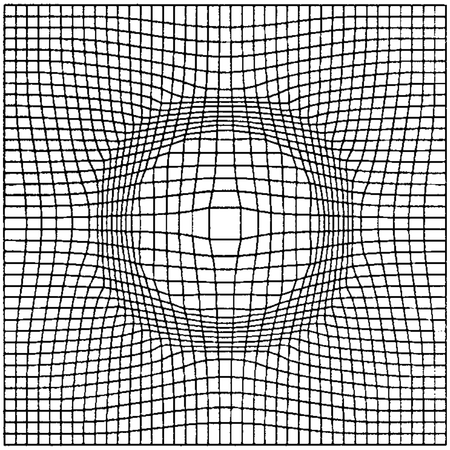

FIG. 4. Adaption to a cylindrical shock front.

airfoil case depicted in Fig. 2, the straight line distance between P_1 and P_2 would traverse through the airfoil. A logical redefinition of this metric for the purpose of computing the Green's function would be to solve the minimal surface problem that yields the shortest path on a (ξ, η) surface between the two points. As an intuitive equivalent, Laplace's equation has been used instead of the above PDE in this work. First, it can be shown [14] that the minimal surface equation transforms into canonial form, a system of Laplace equations, through a bilinear transformation. Second, the Laplace solution for a given domain has already been obtained so no additional effort needs to be expended to find a suitable distance. Figure 3 shows a grid around an airfoil that has been adapted using this scheme.

Great importance is placed on the role of the Laplace solution, ξ_L and η_L, in the adaption process. Thus, care must be exercised to preserve the accuracy of the Laplace solution as the grid points move in time. When a point is moved to a new location, a bilinear interpolation is used within the host cell, preserving second-order accuracy. The same interpolation functions are used to move the Laplace solution for ξ_L and η_L to the new point. Thus, second-order accuracy can be retained in transferring the flow properties to the new point with very little additional effort.

After the Green's functions are computed, they are combined with the weight functions as the source terms P and Q to produce the final adapted grid:

$$\begin{aligned}\xi(x, y) &= \sum_i \sum_j P(x, y)\xi_G(x, y; x_{ij}, y_{ij}) \operatorname{Vol}(i, j)\\ \eta(x, y) &= \sum_i \sum_j Q(x, y)\xi_G(x, y; x_{ij}, y_{ij}) \operatorname{Vol}(i, j).\end{aligned} \tag{25}$$

The approximate Green's functions are computed as the solution proceeds.

Grids generated with approximate Green's functions are shown in Figs. 4 and 5. It is apparent that these grids are smooth and well adapted to the discontinuities in the weight function. Figures 6 and 7 show the effect of changing r_0. Consistent with the elliptic nature of the generating system, larger values of r_0 give smoother results. The most important result, however, is the comparison of CPU time with the standard Thompson scheme shown in Fig. 8. Clearly the time taken by the relaxation method is of the same order as that necessary for the approximate Green's function method.

This method was applied to three dimensions with success and results of the computations are shown in Fig. 9 and Fig. 10. Grids generated with this method are smoother in three dimensions due to the fact that Green's functions

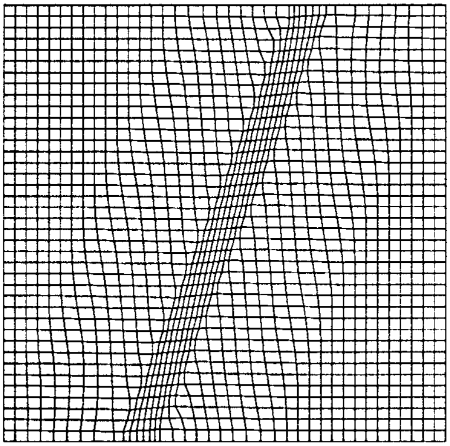

FIG. 5. A shock-like discontinuity.

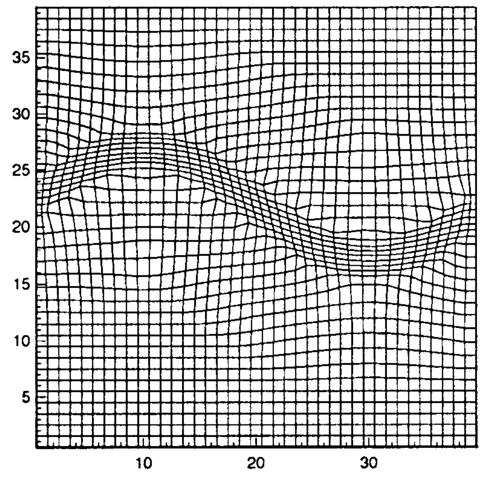

FIG. 6. Adaption to a sine wave with $r_0 = 100$ on the scale shown.

behave like $1/r$. In two dimensions, the $\log(r)$ behavior has a much larger slope when r changes, leading to larger errors in the approximate Green's functions.

THE BOUNDARY ELEMENT METHOD (BEM)

The boundary element method (BEM) is an interesting extention of the Green's function method in grid generation. This technique requires discretization of the surface instead of the volume. Application of the BEM to the Poisson equation is presented below and more complete details may be found in Ref. [15].

Consider the equation

$$\nabla^2 u = b(x, y), \tag{26}$$

where on the boundary, $\Gamma = \Gamma_1 + \Gamma_2$, the boundary conditions are

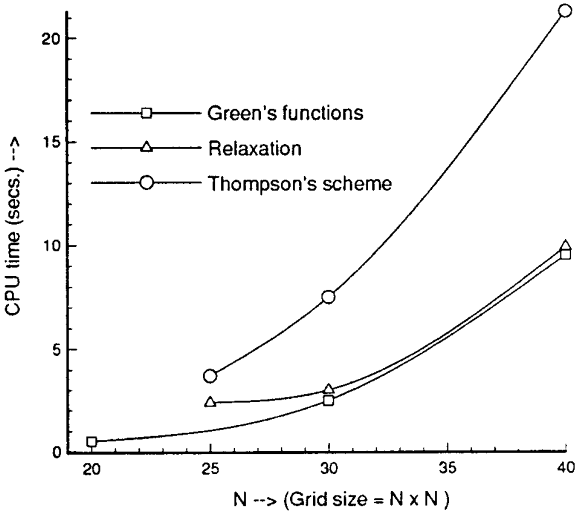

FIG. 8. CPU time comparisons for adaption to a sine wave.

$$\begin{aligned} u &= \quad \overline{u} \qquad \text{on } \Gamma_1 \\ \frac{\partial u}{\partial n} &= q = \overline{q} \quad \text{on } \Gamma_2. \end{aligned} \tag{27}$$

Define the residual (error) in the Poisson computation as

$$R = \nabla^2 u - b, \quad R_1 = u - \overline{u}, \quad R_2 = q - \overline{q}. \tag{28}$$

With these definitions, the basic integral equation may be written

$$\int_\Omega (Ru^*)\, d\Omega = \int_{\Gamma_2} R_2 u^*\, d\Gamma - \int_{\Gamma_1} R_1 q^*\, d\Gamma \tag{29}$$

with $q^* = \partial u^*/\partial n$. The quantity, u^*, represents the funda-

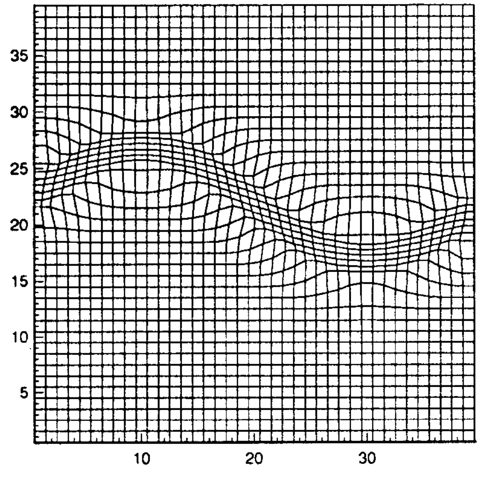

FIG. 7. Grid in Fig. 6 with $r_0 = 12$.

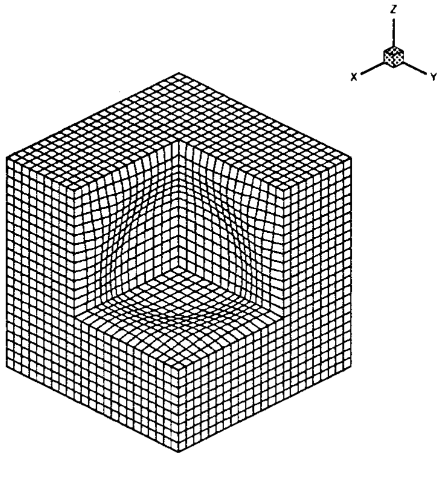

FIG. 9. Adaption to a spherical front.

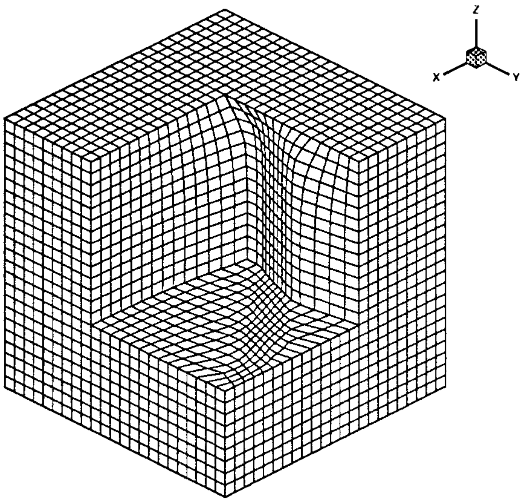

FIG. 10. Adaption to a 3D shock.

mental solution at a point due to a unit impulse at the ith point. Integration by parts yields

$$\int_{\omega} (\nabla^2 u^*) u \, d\Omega - \int_{\Omega} b u^* \, d\Gamma = -\int_{\Gamma_2} \bar{q} u^* \, d\Gamma - \int_{\Gamma_1} q u^* \, d\Gamma + \int_{\Gamma_2} u q^* \, d\Gamma + \int_{\Gamma_1} \bar{u} q^* \, d\Gamma. \tag{30}$$

When all of the boundary terms are grouped together and the effects of P point sources of strength Q^k within the domain are included, the final form of the governing equation is obtained in the form

$$c^i u^i + \int_{\Gamma} u q^* \, d\Gamma + \int_{\Omega} b u^* \, d\Omega + \sum_{k=1}^{P} (Q^k u^{*k} = \int_{\Gamma} q u^* \, d\Gamma. \tag{31}$$

In this expression, $c^i = \theta/2\pi$, where θ is the angle made by the boundary at the ith node. For smooth boundaries, the correct value of θ is 0.5. The volume integral is avoided by means of several simplifying assumptions. The easiest way to simplify this equation is to assume that the right-hand side source term in Eq. (26) is harmonic, i.e.,

$$\nabla^2 b = 0. \tag{32}$$

This may be accomplished by means of the transformation

$$u^* = \nabla^2 v^* \tag{33}$$

and writing Green's identity in the form

$$\int_{\Omega} \Bigl(b \, \nabla^2 v^* - v^* \, \nabla^2 b) \, d\Omega = \int_{\Gamma} b \frac{\partial v^*}{\partial n} - v^* \frac{\partial b}{\partial n} \Bigr) \, d\Gamma. \tag{34}$$

If b is harmonic, this expression reduces to

$$\int_{\Omega} b u^* \, d\Omega = \int_{\Gamma} \left(b \frac{\partial v^*}{\partial n} - v^* \frac{\partial b}{\partial n} \right) d\Gamma, \tag{35}$$

resulting in a pure boundary value problem. The function, v^*, will be the fundamental solution to the biharmonic equation

$$\nabla^2 u^* = \nabla^2(\nabla^2 v^*) = \nabla^4 v^* = -\delta_{\text{Dirac}}. \tag{36}$$

The v^* function for two and three dimensions is given by

$$v^*_{2d} = \frac{r^2}{8\pi} \left[\log \left(\frac{1}{r} \right) + 1 \right], \quad v^*_{3d} = \frac{r}{8\pi}. \tag{37}$$

A further simplification of the solution procedure may be achieved if the source term in the Poisson equation is assumed to be harmonic. As an example, consider adaption in the unit square with clustering desired at the lower boundary. This is easily achieved using a wide variety of choices for the source functions. In this case, $P = 0$ to cluster to the lower boundary and Q may be written as

$$Q = \frac{1}{w} \frac{dw}{dy}, \quad w = 1 + A|\nabla u|^2, \quad u = \tanh(y/c) \tag{38}$$

$$Q = Ae^{By}. \tag{39}$$

Clearly, neither of these functions is harmonic. The simplest harmonic function that will work is the straight line variation

$$Q = A - By.$$

When the domain is not the simple unit square, the source function can be transferred to the computational space of the Laplacian coordinates, (ξ_L, η_L) (also harmonic functions), where it must be remembered that these functions are available during the course of the current computation. A linear function of the Laplacian coordinates is harmonic. If

$$P = A\xi_L + B\eta_L + C, \quad \nabla^2 P = a \, \nabla^2 \xi_L + B \, \nabla^2 \eta_L, \tag{40}$$

where A, B, and C are constants, $\nabla^2 P = 0$, making P harmonic. In general, a function that is harmonic in the (x, y) plane can be transformed into computational space with a restriction on the Laplacian coordinates. Consider a harmonic function P. After simplification, the Laplacian of P can be written

$$\nabla_x^2 P(\xi_L, \eta_L) = P_{\xi_L \xi_L}(\nabla \xi_L \cdot \nabla \xi_L) + P_{\eta_L \eta_L}(\nabla \eta_L \cdot \nabla \eta_L) + 2P_{\xi_L \eta_L}(\nabla \xi_L \cdot \nabla \eta_L). \tag{41}$$

If we also impose the restriction that the functions ξ_L and η_L satisfy the Cauchy–Riemann conditions, then

$$\frac{\partial \xi_L}{\partial x} \quad \frac{\partial \eta_L}{\partial y} \tag{42}$$

$$\frac{\partial \xi_L}{\partial y} = -\frac{\partial \eta_L}{\partial x} \tag{43}$$

and Eq. (40) is substantially simplified. These simplifications include

$$\nabla \xi_L \cdot \nabla \eta_L = 0$$
$$\nabla \xi_L \cdot \nabla \xi_L = \nabla \eta_L \cdot \nabla \eta_L$$

and, since

$$P_{\xi_L \xi_L} + P_{\eta_L \eta_L} = 0,$$

we obtain

$$\nabla_x^2 P = 0. \tag{44}$$

By transferring the quantities P and Q to the (x, y) domain, the adaption is also transferred to the relevant regions in the domain.

Implementation and Results

The BEM can be applied in the present form to several important cases in a non-iterative format. Consider the boundary integral formulation of Eq. (31). This expression can be written

$$c^i \xi^i = \xi_L - \int_\Gamma \left(b \frac{\partial \xi^*}{\partial n} - \xi^* \frac{\partial b}{\partial n} \right) d\Gamma - \sum_{l=1}^{P} (Q^l \xi^{*l}). \tag{45}$$

The term, ξ_L, is the Laplace solution in the domain, as can be seen by setting the source term to zero in the integral relation. The integral represents the influence of a harmonic source function from the boundary and the summation is the effect of discrete internal sources. If the source function, b, is known as a function of x and y, closed form expressions can be obtained for the integral in Eq. (45). It must be remembered that the distance metric being used for the Green's function computation is the Laplacian metric and not the Euclidian metric. This implies that the adaption is being performed in the $\xi_L - \eta_L$ space.

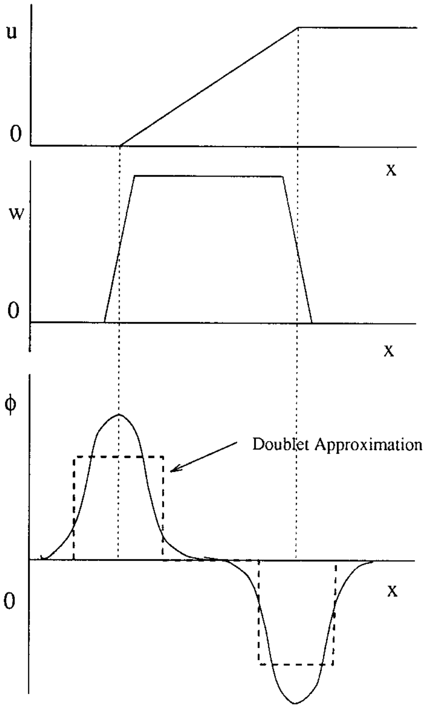

FIG. 11. The doublet approximation to ϕ and ψ.

A simplification of the current problem can be made by setting the source function equal to zero leaving only the influence of discrete sources.

Internal Sources

Provisions exist for including isolated internal source terms in the BEM formulation (Eq. (31)). Suppose that clustering is desired at a given point in the region. For a one-dimensional case, this requires definition of a source function, ϕ, that depends on the second derivative of a weight function. With this interpretation, the source function is written

$$\phi = \frac{w_{\xi\xi}}{w},$$

where the weight function is defined as

$$w = 1 + A|\nabla u|^2.$$

A definition of u and w in this manner produces an oscillatory ϕ distribution. Typical behavior of this function is shown in Fig. 11. It is important to note that ϕ exhibits a sign change in order to attract grid points from both directions in a given region. A doublet or dipole (Fig. 11)

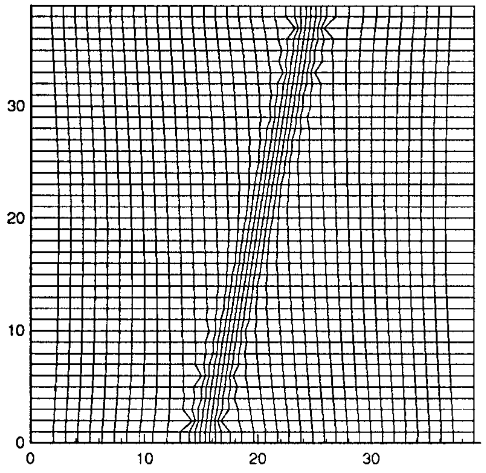

FIG. 12. Adaption using 20 doublets along a shock wave.

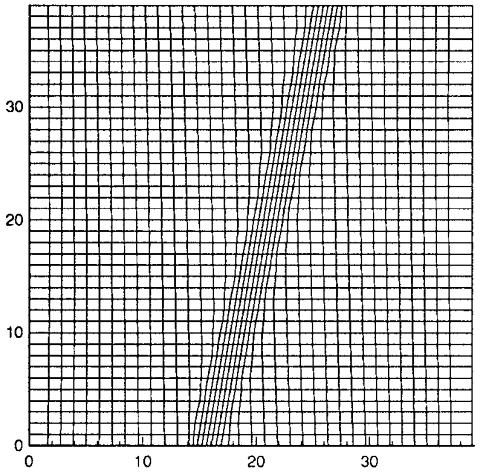

FIG. 13. Grid obtained using a continuous vortex filament.

has the correct behavior and may be used to attract points to a given location. The term "doublet" is used loosely to describe a system of positive–negative singularities. In generating a grid, the doublet distribution is determined from the gradient information as a first step in the grid generation process and stored for future use. In order to orient the doublet along the coordinate lines, the doublet direction is chosen to be aligned with $\nabla\xi$ for computing ξ and $\nabla\eta$ for computing η. Values of ξ and η at every point may be evaluated with Eq. (31) and an interpolation step yields the final grid. The adaption to a shock-like discontinuity has been carried out in a square domain using 20 doublets. The resulting grid is shown in Fig. 12. While the grid points are seen to cluster near the desired region, there are spurious wiggles in the grid lines that violate the smoothness requirements. This is attributed to the discrete placement of the doublets. In order to avoid this situation, we may replace these doublets with continuous vortex filaments of opposing strengths that can be placed parallel to each other in order to attract points smoothly from either direction. The grid resulting from using continuous vortex filaments is shown in Fig. 13.

The influence on a point due to a straight "vortex" filament that stretches between points 1 and 2 that is a perpendicular distance h away at distance $r(\xi_L, \eta_L)$ is easily computed. If the influence of a small segment, ds, of the line is given by

$$d\xi = \kappa \log\left(\frac{r}{r_o}\right) ds$$

then the net influence at a given point may be represented as

$$\Delta\xi = \kappa\left[s\log(r) - s + h\cos\left(\frac{h}{r}\right) - s\log\left(r_o\right)\right]_1^2.$$

Using discrete doublets produces interesting results. Figure 14 depicts the effect of placing 30 doublets oriented in the horizontal direction along a sine-wave-like discontinuity. Figure 15 shows the doublets inclined to the normal to the sinusoidal discontinuity. If such an adaption were to be achieved using vortex filaments, it would require that the sine wave be broken up into piecewise linear segments along which vortex filaments can be placed.

The next case of interest is that of flow over an NACA 0012 airfoil where adaption is desired near the shock to reduce the inviscid shock mesh spacing. The grid generation routine searches the domain during the solution procedure to locate the region where the pressure gradient is large. A decision is then made to use either discrete vortex doublets or continuous filaments. In the present case, the method is programmed to recognize a single straight line starting from the boundary and terminating in the flow

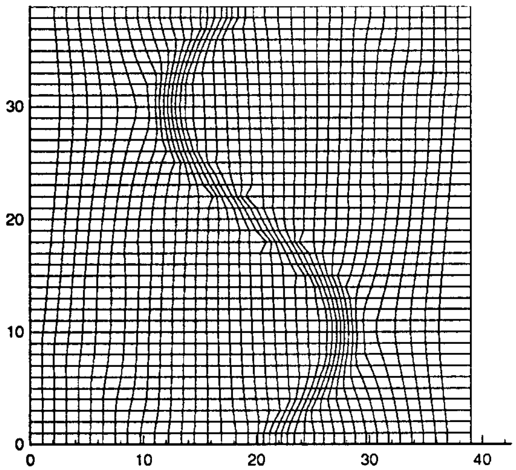

FIG. 14. Distribution of 30 doublets.

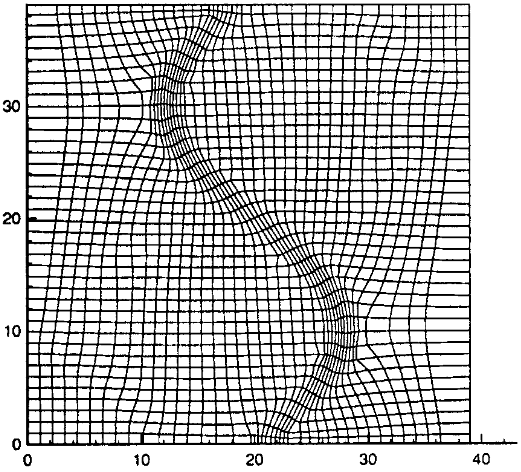

FIG. 15. Effect of orienting the doublets on orthogonality.

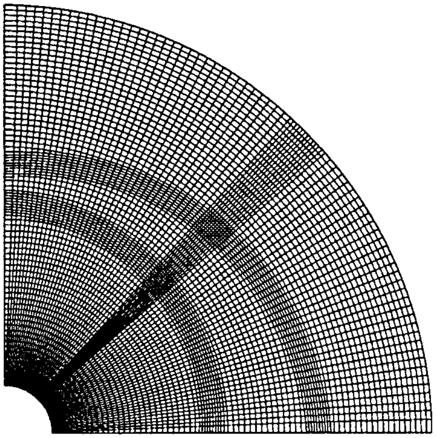

FIG. 17. Use of multiple filaments in a domain.

domain that would approximate the shock location. Since the transonic shock ends in the flow-field, the vortex filaments must end within the flow-field. This causes the grid lines to possess large curvature near the regions where the vortex filament must end. In order to avoid this problem, the filaments are chosen as convergent lines that meet near the point where the shock ends. The grid obtained for this case is shown in Fig. 16.

An arbitrary number of vortex filaments can be placed in the flow domain. An example of three filaments is represented in Fig. 17. Given the capability of vortex lines to control adaption in an arbitrary problem, it remains to demonstrate the utility of adding continuous source functions to the Poisson solver. In order to fit into the BEM format, we require these source functions to be harmonic, resulting in the biharmonic equation for grid generation. However, if the source term is harmonic, an extremum cannot occur within the domain due to the max–min property for harmonic functions. Such a function cannot be used to obtain grid adaption interior to the domain. At the same time, these functions can serve two important purposes, adaption to the boundary and smoothing an adapted grid.

A linear or piecewise linear variation of a source function within the (ξ_l, η_L) space gives rise to adaption that originates at the boundaries and goes all the way into the domain. Figure 18 shows adaption to the $\eta = 0$ boundary for the NACA 0012 airfoil by using a linear source function that vanishes at the upper boundary and reaches some small positive value at the lower boundary.

The evaluation of the boundary integral is carried out, for harmonic source functions, in several steps. This computation requires the gradient of the function, v^*, normal

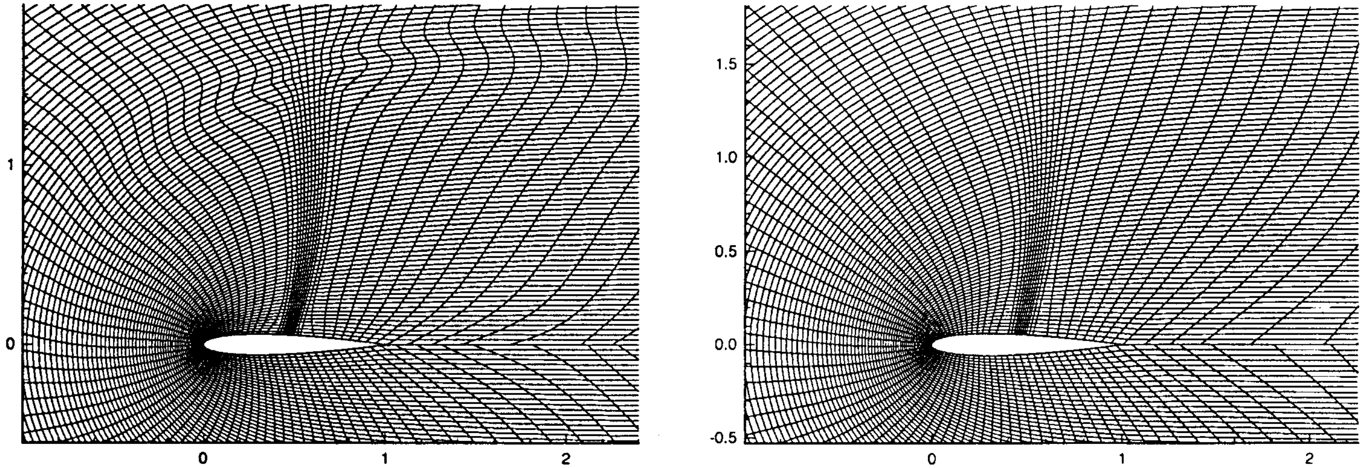

FIG. 16. Parallel and convergent vortex filaments.

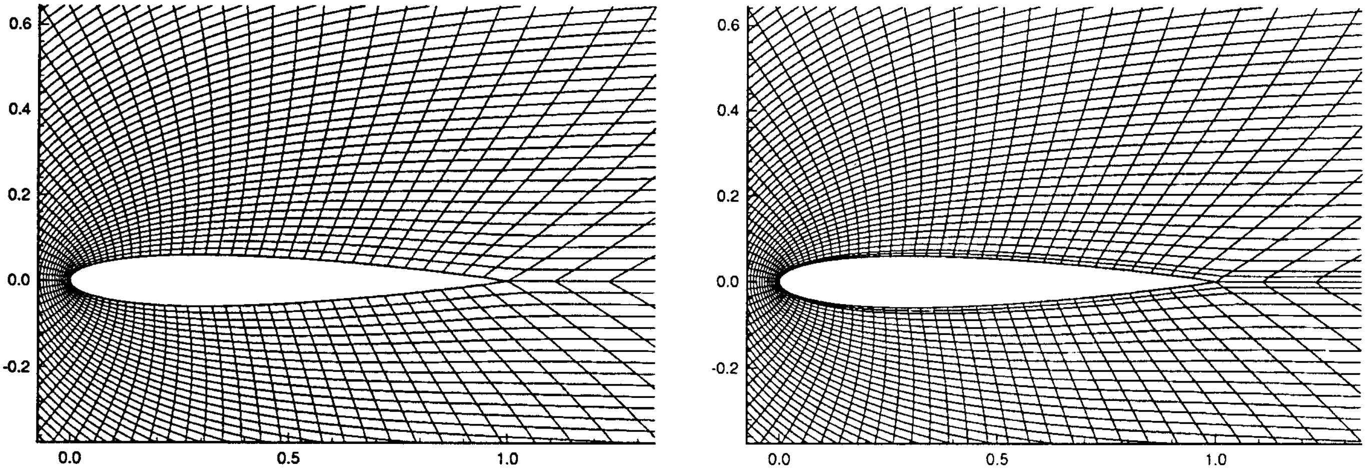

FIG. 18. Using a vortex line for boundary treatment.

to the boundary and the gradient of b normal to the boundary. The first of these, ∇v^*, is given by

$$\nabla v^* = \left[\frac{r}{4\pi}\left(\log\frac{1}{r}+1\right)-\frac{1}{8\pi r}\right)\mathbf{e}_r,$$

where $\mathbf{e}_r$ is the unit vector in the radial direction along the line joining the point at which the computation is being made to a point on the boundary. The determination of ∇b is not as straightforward. The problem is simple if a function $b(\xi_L, \eta_L)$ is given. As noted earlier, such a specification can be useful in many cases such as adaption to a boundary. In general, a harmonic source function cannot be used for internal adaption. One must then resort to using this for the second function mentioned above, namely, smoothing an adapted grid. Consider the grid depicted in Fig. 17. The grid lines have been clustered near the linear region shown using vortex filaments. Such a grid can potentially be smoothed using a continuous source function that is harmonic.

It is interesting to examine the construction of such a continuous harmonic source function. Since the function b is harmonic, the specification of its value on the boundary completely determines the internal distribution. In Fig. 17, we observe that the only region where b is required is near the transonic shock. Thus, we can set the value of b to zero everywhere else on the boundary. Since clustering is required near the shock, we let b vary linearly from zero to a small positive value k at the shock, where it must change sign to $-k$ in order to attract points from both directions. It then drops linearly to zero at the outer edge as shown in Fig. 19.

With this boundary distribution of b, we can obtain an estimate of the normal derivative at the boundaries using the Fourier series solution of Laplace's equation. The following expressions may be used:

$$b(\xi_L, \eta_L) = \frac{2}{\xi_{\max}}\sum_1^\infty \frac{\sinh[(n\pi/\xi_{\max})\eta_l]}{\sinh[\eta\pi\eta_{\max}/\xi_{\max}]}$$
$$\sin\frac{\pi n\xi_L}{L}\int_0^L f(\xi_L)\sin\frac{n\pi\xi_L}{L}\,dx$$
$$\frac{\partial b}{\partial\eta_L}\Big|_{\eta_L=\eta_{\max}} = \frac{2n\pi}{\xi_{\max}^2}\sum_1^\infty \frac{\cosh[n\pi\eta_{\max}/\xi_{\max}]}{\sinh[\eta\pi\eta_{\max}/\xi_{\max}]}$$
$$\sin\frac{n\pi\xi_L}{L}\int_0^L f(\xi_L)\sin\frac{n\pi\xi_L}{L}\,dx.$$

This expression has been used to smooth the grid shown in Fig. 17. The function $f(\xi_l)$, is identical to the boundary distribution of b discussed above. The Fourier series has been truncated at three terms in the computation. The

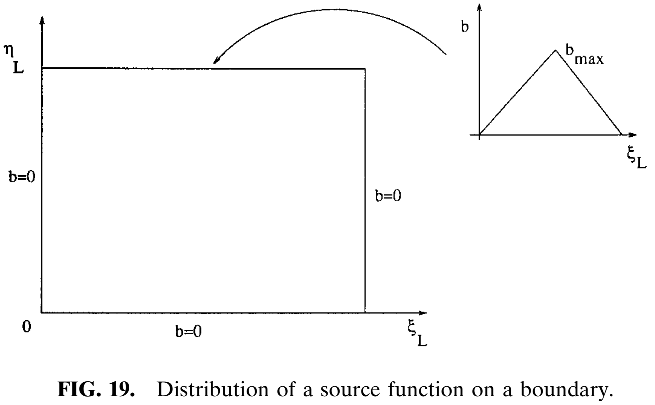

FIG. 19. Distribution of a source function on a boundary.

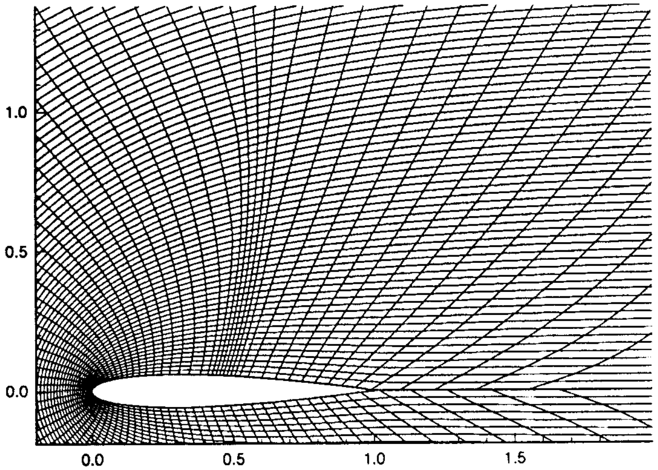

FIG. 20. Grid in Fig. 16 after adding continuous source terms.

smooth grid has been plotted in Fig. 20, showing the enhanced smoothness. The grid line spacing varies more uniformly away from the discontinuity. The strengths of the vortex lines and the source functions may be varied in order to control the amount of adaption and smoothness present in the grid.

CONCLUSIONS

The BEM has been observed to fit quite naturally into the grid generation scenario. The use of doublets to generate grids has the potential to reduce the work required for adaption by an order of magnitude and actually make the solution methodology close to an order-N process. This important reduction in effort using the BEM formulation produces smooth, oscillation-free grids when continuous vortex lines and sufficiently high doublet densities are used.

The BEM is one way to avoid evaluation of double summations by making judicious use of the information pertaining to flow gradients. Grid computation can be modified near the regions where there is little or no activity. Implicit approximations to the forcing functions P and Q can possibly be obtained, thereby permitting integration of the Poisson equation, resulting in a closed-form solution. If piecewise harmonic functions are included, boundary-point-dependent adaption is easily achieved.

The large number of parameters in the formulation can be manipulated to obtain orthogonality and other grid properties of practical interest. In addition, the interpolator can serve as another degree of freedom in grid control. Further study is necessary to understand the best way of including the effects of orthogonality and smoothness in this scheme. Linearity decouples the Poisson equations totally so that the method has no dependence on the complexity of the problem. CPU time should increase linearly with each additional dimension or increasing grid size while any non-linear scheme proceeds along some higher exponent of this CPU time.

REFERENCES

1. A. Winslow, *J. Comput. Phys.* **1,** 149 (1966).
2. J. F. Thompson, F. C. Thames, and C. M. Mastin, *J. Comput. Phys.* **18,** 652 (1980).
3. W. D. Barfield, *J. Comput. Phys.* **6,** 417 (1970).
4. A. A. Amsden and C. W. Hirt, *J. Comput. Phys.* **11,** 348 (1973).
5. S. K. Godunov and G. P. Prokopov, *J. Comput. Phys.* **12,** 182 (1972).
6. P. D. Thomas and J. F. Middlecoff, *AIAA J.* **18,** No. 6, 652 (1980).
7. M. M. Rai and D. A. Anderson, *J. Comput. Phys.* **43,** 327 (1981).
8. J. U. Brackbill and J. S. Saltzman, *J. Comput. Phys.* **46,** 342 (1982).
9. D. A. Anderson, *Appl. Math. Comput.* **24,** 211 (1987).
10. D. A. Anderson, *Appl. Math. Comput.* **35,** 209 (1990).
11. G. Liao and D. A. Anderson, *Appl. Anal.* **44,** 285 (1992).
12. J. Moser, *Trans. Am. Math. Soc.* **120,** 286 (1965).
13. R. Munipalli and D. A. Anderson, "Linearized Adaption in Structured Grids," AIAA Paper 95-08562, in *The 33rd Aerospace Sciences Meeting, Reno, NV, Jan. 1995.*
14. P. R. Garabedian, *Partial Differential Equations,* 2nd ed. (Chelsea, New York, 1986).
15. C. A. Brebbia and J. Dominguez, *Boundary Elements—An Introductory Course* (Comput. Mech., Southampton, 1989).