JOURNAL OF COMPUTATIONAL PHYSICS 115, 440–454 (1994)

Curvature Flow and Entropy Conditions Applied to Grid Generation

J. A. SETHIAN*

Department of Mathematics, University of California, Berkeley, California 94720

Received April 11, 1994

We describe a numerical technique to generate logically rectangular body-fitted interior and exterior grids. The technique is based on solving a Hamilton–Jacobi-type equation for a propagating level set function, using techniques borrowed from hyperbolic conservation laws. Coordinate grid lines are kept smooth through curvature terms which regularize the equation of motion, and upwind difference schemes which satisfy the correct entropy conditions of front propagation. The resulting algorithm can be used to generate two- and three-dimensional interior and exterior grids around reasonably complex bodies which may contain sharp corners and significant variations in curvature. The technique may also be easily extended to problems of boundary-fitted moving grids. © 1994 Academic Press, Inc.

1. INTRODUCTION

In this paper, we present a numerical technique to generate body-fitted logically rectangular grids. The technique hinges on viewing the boundary of the body as a propagating front. The front is then allowed to propagate with a speed law that ensures that it will smoothly evolve from the body in such a way that the position of the front yields one set of coordinate lines. This smooth propagation is achieved by viewing the front as the zero level set of a higher dimensional function whose equation of motion is described by an initial value partial differential equation. This equation of motion is approximated by techniques borrowed from the numerical solution of hyperbolic conservation laws which guarantee that the correct entropy-satisfying solution will be produced. The evolving front is thus a hypersurface, i.e., a curve in two dimensions and an evolving surface in three space dimensions.

The resulting algorithm can be used to generate two and three dimensional interior and exterior grids around reasonably complex bodies which may contain sharp corners and significant variations in curvature. This evolution equation for the grid can also be modified to produce a non-uniform solution-adaptive mesh. The technique may also be easily extended to problems of boundary-fitted moving grids.

* Supported in part by the Applied Mathematics Subprogram of the Office of Energy Research under Contract DE-AC03-76SF00098, and the National Science Foundation and DARPA under Grant DMS-8919074.

This technique produces a logically rectangular grid which represents a mapping from a cartesian grid on the unit square to the domain space. More specifically, in two space dimensions the technique produces a discrete function $\Omega(s, t)$: $[0, 1]x[0, 1] \rightarrow R^2$, where $\Omega(s, t = 0)$ is mapped onto the body itself and time t corresponds to evolution away from the body. The grids produced by this technique may either be used directly in the computational simulation, or taken as a first stage in some other grid solver. In such a case, one may chose to modify the grid under a variety of well-developed and more traditional grid smoothers, such as those that employ elliptic or diffusion operators to readjust grid nodes.

The subject of grid generation is vast, with a variety of highly evolved techniques. The reader is referred to the recent book by Knupp and Steinberg [23] and a SIAM monograph [4], as well as [10, 33]. The main goal of grid generation in our context is to produce a body-fitted, logically rectangular grid which balances the competing demands of uniformity in cell sizes and mesh orthogonality. Broadly speaking, the techniques for logically rectangular grids include generation of grids by hand, algebraic and approximation methods, elliptic techniques which solve a steady elliptic partial differential equation, hyperbolic techniques which advance a front away from the body in such a way as to produce an appropriate set of grid lines as the front propagates outwards, and variational methods.

The technique presented here roughly falls in the category of a hyperbolic solver. However, as will be discussed in detail in the next section, we avoid the difficulties of shock formation and colliding characteristics by solving the correct evolution equation for an advancing front which naturally avoids these singularities. User intervention is kept to a minimum; for the most part, grids are generated automatically without the need to adjust parameters.

We use these grid generation techniques to design grids around a variety of bodies, including interior and exterior grids around convex bodies, airfoils, sharp cornered shapes, bodies with extending limbs and cavities, and body-fitted grids around moving interfaces.

The outline of this paper is as follows. In Section 1, we discuss the basic ideas behind this entropy-based level set approach to advancing fronts. In Section 2, we show how these techniques can be applied to grid generation by constructing appropriate

0021-9991/94 $6.00

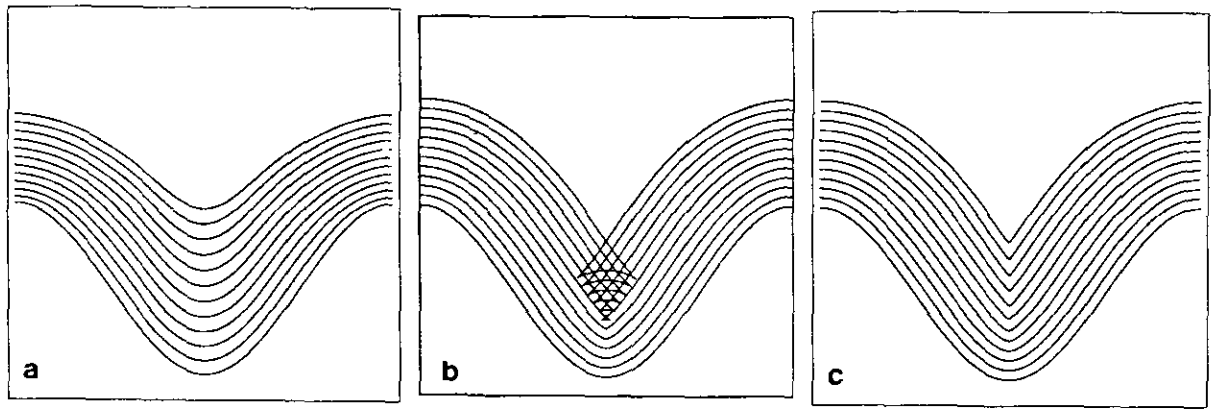

FIG. 1. Propagating cosine curve. (a) $F = 1 - 0.25\ K$; (b) swallowtail, $F = 1.0$; (c) entropy satisfying, $F = 1.0$.

evolution equations and techniques for generating transverse lines. In Sections 3 and 4, we discuss some details, including the generation of farfield conditions, techniques to moving interface-fitted grids, and extensions to three space dimensions. In Section 5, we give examples of grids generated using our approach, and conclude in Section 6 with a discussion of future work.

2. NUMERICAL ALGORITHMS FOR PROPAGATING FRONTS

2.1. *Entropy Conditions and Curvature*

The fundamental aspects of front propagation in our context can be illustrated as follows. Let $\gamma(0)$ be a smooth, closed initial curve in R^2, and let $\gamma(t)$ be the one-parameter family of curves generated by moving $\gamma(0)$ along its normal vector field with speed $F(K)$. Here, $F(K)$ is a given scalar function of the curvature K. Thus, $n \cdot x_t = F(K)$, where x is the position vector of the curve, t is time, and n is the unit normal to the curve.

Consider a speed function of the form $1 - \varepsilon K$, where ε is a constant. An evolution equation for the curvature K, see [28], is given by

$$K_t = \varepsilon K_{\alpha\alpha} + \varepsilon K^3 - K^2 \tag{1}$$

where we have taken the second derivative of the curvature K with respect to arclength α. This is a reaction-diffusion equation; the drive toward singularities due to the reaction term $(\varepsilon K^3 - K^2)$ is balanced by the smoothing effect of the diffusion term $(\varepsilon K_{\alpha\alpha})$. Indeed, with $\varepsilon = 0$, we have a pure reaction equation $K_t = -K^2$. In this case, the solution is $K(s, t) = K(s, 0)/(1 + tK(s, 0))$, which is singular in finite t if the initial curvature is anywhere negative. Thus, corners can form in the moving curve when $\varepsilon = 0$.

As an example, consider the periodic initial cosine curve

$$\gamma(0) = (-s, [1 + \cos 2\pi s]/2) \tag{2}$$

propagating with speed $F(K) = 1 - \varepsilon K$, $\varepsilon > 0$. As the front moves, the troughs at $s = n + 1/2$, $n = 0, \pm 1, \pm 2, \ldots$, are sharpened by the negative reaction term (because $K < 0$ at such points) and smoothed by the positive diffusion term (see Fig. 1a). For $\varepsilon > 0$, it can be shown (see [28, 26]) that the moving front stays C^∞.

On the other hand, for $\varepsilon = 0$, the front develops a sharp corner in finite time as discussed above. In general, it is not clear how to construct the normal at the corner and continue the evolution, since the derivative is not defined there. One possibility is the "swallowtail" solution formed by letting the front pass through itself (see Fig. 1b). However, from a geometrical argument it seems clear that the front at time t should consist of only the set of all points located a distance t from the initial curve. (This is known as the Huygens principle construction, see [28]). Roughly speaking, we want to remove the "tail" from the "swallowtail." In Fig. 1c, we show this alternate weak solution. Another way to characterize this weak solution is through the following "entropy condition" (see [28]): If the front is viewed as a burning flame, then *once a particle is burnt it stays burnt.* Careful adherence to this stipulation produces the Huygens principle construction. Furthermore, this physically reasonable weak solution is the formal limit of the smooth solutions $\varepsilon > 0$ as the curvature term vanishes (see [26, 28]).

The key to constructing numerical schemes which adhere to this entropy condition comes from the link between propagating fronts and hyperbolic conservation laws. Consider the initial front given by the graph of $f(x)$, with f and f' periodic on $[0, 1]$, and suppose that the propagating front remains a function for all time. Let ϕ be the height of the propagating function at time t, thus $\phi(x, 0) = f(x)$. The normal at (x, ϕ) is $(1, \phi_x)$, and the equation of motion becomes $\phi_t = F(K)(1 + \phi_x^2)^{1/2}$. Using

the speed function $F(K) = 1 - \varepsilon K$ and the formula $K = -\phi_{xx}/(1 + \phi_x^2)^3/2$, we get

$$\phi_t - (1 + \phi_x^2)^{1/2} = \varepsilon \frac{\phi_{xx}}{1 + \phi_x^2}. \tag{3}$$

Differentiating both sides of this equation yields an evolution equation for the slope $u = d\phi/dx$ of the propagating front, namely

$$u_t + [-(1 + u^2)^{1/2}]_x = \varepsilon \left[\frac{u_x}{1 + u^2} \right]_x. \tag{4}$$

Thus, the derivative of the Hamilton–Jacobi equation with parabolic right-handside for the changing height ϕ is a viscous hyperbolic conservation law for the propagating slope u (see [30]). Our entropy condition is in fact equivalent to the one for propagating shocks in hyperbolic conservation laws. Thus, we exploit the numerical technology from hyperbolic conservation laws to build consistent, upwind schemes which select the correct entropy conditions. For details, see [26, 29].

Our goal then is to choose an appropriate speed function that yields front motion away from the body that remains smooth for all time, and thus can act to define one set of body-fit coordinate lines. Before doing so, we must extend the above ideas to include propagating fronts which are not easily written as functions. This is the level set idea introduced by Osher and Sethian [26].

2.2. *Level Set Methods*

Given a moving closed hypersurface $\Gamma(t)$, that is, $\Gamma(t = 0)$: $[0, \infty) \to R^N$, we wish to produce an Eulerian formulation for the motion of the hypersurface propagating along its normal direction with speed F, where F can be a function of various arguments, including the curvature, normal direction, etc. The main idea is to embed this propagating interface as the zero level set of a higher dimensional function ϕ. Let $\phi(x, t = 0)$, where $x \in R^N$ is defined by

$$\phi(x, t = 0) = \pm d, \tag{5}$$

where d is the distance from x to $\Gamma(t = 0)$ and the plus (minus) sign is chosen if the point x is outside (inside) the initial hypersurface $\Gamma(t = 0)$. Thus, we have an initial function $\phi(x, t = 0)$: $R^N \to R$ with the property that

$$\Gamma(t = 0) = (x | \phi(x, t = 0) = 0). \tag{6}$$

Our goal now is to produce an equation for the evolving function $\phi(x, t)$ which contains the embedded motion of $\Gamma(t)$ as the level set $\phi = 0$. Here, we follow the derivation presented in [25]. Let $(x(t), t) \in [0, \infty)$, be the path of a point on the propagating front. That is, $x(t = 0)$ is a point on the initial front $\Gamma(t = 0)$, and $x_t = F(x(t))$ with the vector x_t normal to the front at $x(t)$. Since the evolving function ϕ is always zero on the propagating hypersurface, we must have

$$\phi(x(t), t) = 0. \tag{7}$$

By the chain rule,

$$\phi_t + \sum_{i=1}^{N} \phi_{x_i} x_{i_t} = 0, \tag{8}$$

where x_i is the ith component of x. Let $(u_1, u_2, \ldots, u_N) = (x_{1_t}, x_{2_t}, \ldots, x_{N_t})$. Since

$$\sum_{i=1}^{N} \phi_{x_i} x_{i_t} = (\phi_{x_1}, \phi_{x_2}, \ldots, \phi_{x_N}) \cdot (u_1, u_2, \ldots u_N) = F(x(t))|\nabla \phi| \tag{9}$$

we then have the evolution equation for ϕ, namely

$$\phi_t + F|\nabla \phi| = 0 \tag{10}$$

$$\phi(x, t = 0) \text{ given.} \tag{11}$$

We refer to this as a Hamilton–Jacobi- "type" equation because, for certain forms of the speed function F, we obtain the standard Hamilton–Jacobi equation.

There are four major advantages to this Eulerian Hamilton–Jacobi formulation. The first is that the evolving function $\phi(x, t)$ always remains a function as long as F is smooth. However, the level surface $\phi = 0$, and hence the propagating hypersurface $\Gamma(t)$, may change topology, break, merge, and form sharp corners as the function ϕ evolves; see [26].

The second major advantage of this Eulerian formulation concerns numerical approximation. Because $\phi(x, t)$ remains a function as it evolves, we may use a discrete grid in the domain of x and substitute finite difference approximations for the spatial and temporal derivatives. For example, using a uniform mesh of spacing h, with grid nodes ij, and employing the standard notation that ϕ_{ij}^n is the approximation to the solution $\phi(ih, jh, n\,\Delta t)$, where Δt is the time step, we might write

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} + (F)(\nabla_{ij} \phi_{ij}^n) = 0. \tag{12}$$

Here, we have used forward differences in time, and let $\nabla_{ij} \phi_{ij}^n$ be some appropriate finite difference operator for the spatial derivative. As discussed above, the correct entropy-satisfying approximation to the difference operator comes from exploiting the technology of hyperbolic conservation laws. Following [26], we use a modification of an Enquist–Osher schemes [11]. That is, given a speed function $F(K)$, we update the front by the following scheme. First, separate $F(K)$ into a constant advection term F_0 and the remainder $F_1(K)$; that is,

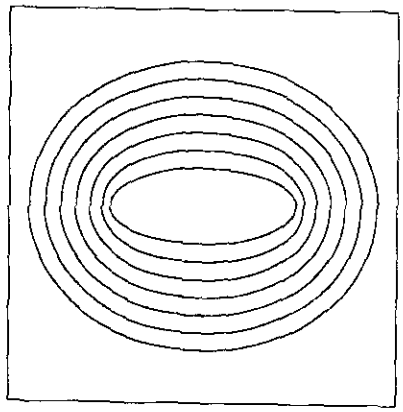

FIG. 2. *Ellipse propagating with speed $F = 1 - 0.25\ K$.*

$$F(K) = F_0 + F_1(K). \tag{13}$$

The advection component F_0 of the speed function is then approximated using upwind schemes, while the remainder is approximated using central differences. In one space dimension, we have

$$\phi_{ij}^{n+1} = \phi_{ij}^n - \Delta t\,[(\max(D_x^-, 0)^2 + \min(D_{x0}^+)^2)^{1/2} - F_1(K)\,\nabla\phi_{ij}^n|. \tag{14}$$

Extension to higher dimensions are straightforward; we use the one-dimensional construction applied to multidimensional problems introduced in [32].

The third major advantage of the above formulation is that intrinsic geometric properties of the front may be easily determined from the level function ϕ. For example, at any point of the front, the normal vector is given by

$$\mathbf{n} = \nabla\phi \tag{15}$$

and the curvature is easily obtained from the divergence of the gradient of the unit normal vector to front, i.e.,

$$K = \nabla\cdot\frac{\nabla\phi}{|\nabla\phi|} = -\frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_x\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}. \tag{16}$$

Finally, the fourth major advantage of the above level set approach is that these are no significant differences in following fronts in three space dimensions. By simply extending the array structures and gradient operators, propagating surfaces are easily handled.

Since its introduction in [26], the above level set approach has been used in a wide collection of problems involving moving interfaces. Some of these applications include the generation of minimal surfaces [7], singularities and geodesics in moving curves and surfaces in [9], flame propagation [27, 34], fluid interfaces [5, 25], and shape reconstruction [24]. Extensions of the basic technique include fast methods in [1] and extensions to triple points in [3]. The fundamental Eulerian perspective presented by this approach has since been adopted in many theoretical analyses of mean curvature flow, in particular, see [2, 6, 8, 12–15, 18].

As an example of the calculation of an interface propagating with curvature dependent speed using this level set approach, in Fig. 2 we show the evolution of an ellipse propagating outwards with speed $F(K) = 1 - 0.25K$.

3. GRID GENERATION

Our basic philosophy is to advance the front away from the body, for either and interior or exterior grid, using a chosen speed function $F(K)$. Throughout this section we shall restrict ourselves to two dimensional grids. At discrete chosen time intervals, zero contours of the level set function ϕ are constructed and serve as one set of grid lines. Transverse lines to these grid lines are then constructed.

3.1. *Generating the Field Lines: Zero Level Set Contours*

3.1.1. *Exterior grids.* We begin by constructing an appropriate speed function to generate body-fitted coordinate lines

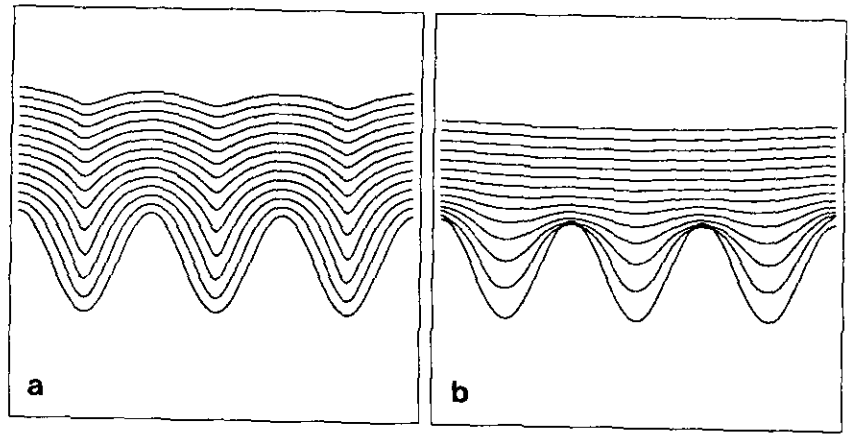

FIG. 3. Propagating triple sine curve: (a) $F = 1 - 0.025\ K$, (b) $F = 1 - 0.25\ K$.

outside a given body. In [28], it is shown that a front propagating with speed $F(K) = 1 - \varepsilon K$, $\varepsilon \geq 0$, must evolve toward a circle and hence yield far-field circular grids.

The rate at which the evolving front becomes circular depends on the choice of ε; the larger the value of ε, the faster the decay in the variation of curvature around the front, see [28]. The choice of ε is controlled by two competing issues, as illustrated in Fig. 3. For ε too small (Fig. 3a), the troughs sharpen up and will result in transverse lines that come too close together. For ε too large (Fig. 3b), parts of the boundary with high values of positive curvature can initially move inwards, and concave parts of the front can move too quickly, both of which are unacceptable for defining grids.

There are several ways to choose ε based on the above discussion. The first is to determine the maximum of the initial curvature, and use that to scale the value of the epsilon. That is,

$$\varepsilon = \max_{\Gamma(t=0)} 1/K. \tag{17}$$

In the case of convex initial boundaries, this is acceptable. However, for nonconvex initial boundaries, such as the double cosine wave shown in Fig. 3, the front can still move inwards under this speed law.

An alternative is to choose a speed function of the form

$$F(K) = e^{-\varepsilon K}. \tag{18}$$

Note that for ε positive, $F(K)$ will always be positive, and hence fronts cannot move inwards. We have performed considerable numerical experimentation with this speed function; however, the results depend quite delicately on the choice of ε.

Instead, we have chosen the following. We choose a positive threshold value $F_{\text{threshold}}$ and set

$$F(K) = \max(1 - \varepsilon K, F_{\text{threshold}}). \tag{19}$$

The choice of ε and $F_{\text{threshold}}$ are determined by the initial transverse line nodal placement as described below in the section on constructing transverse lines.

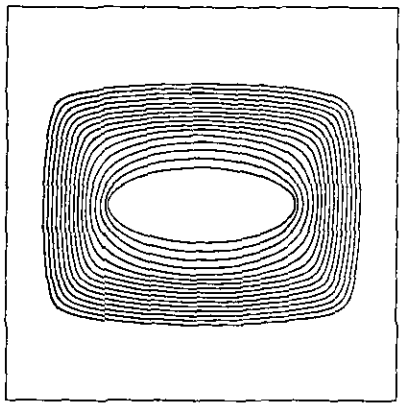

FIG. 4. Ellipse moving under anisotropy.

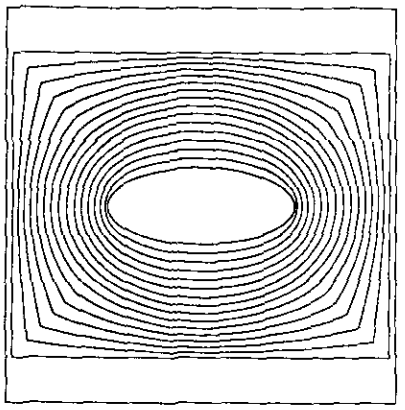

FIG. 5. Ellipse evolving to rectangular boundary.

The above speed function generates exterior grids which are circular away from the body. If a rectangular far-field grid is desired, two straightforward techniques are available. Motivated by the work in crystal growth and dendritic solidification in [32], the first approach is to modify the speed function through a fourfold anisotropy. That is, let the speed function $F(K)$ be defined as

$$F(K) = (\max(1 - \varepsilon K, F_{\text{threshold}}))\,(1 + \alpha_{\text{anisotropy}} \cos(4\theta)), \tag{20}$$

where θ is the angle between the unit normal $\mathbf{n} = \nabla\phi/|\nabla\phi|$ and the positive x-axis. As shown in Fig. 4, this produces field lines which evolve towards a rectangular shape. Here, one must use the approximation to the normal direction given in [32] in order to accurately evaluate normals at the sharpening corners.

An alternative approach, which may be preferable in some situations, comes from a two-stage process. One first starts with the standard speed function provided in Eq. (19). At some user-determined point in the flow, the front is then simply carried through linear extrapolation outwards towards a larger rectangular of chosen size (see Fig. 5). This approach works well if one wants a final exterior rectangular grid that is at least a half characteristic body diameter away from the initial front; that is, time must be allowed for the initial body to evolve into a circular shape before the extrapolation function is initiated.

3.1.2. *Periodic grids.* Periodic grids are obtained in a straightforward manner. Given an initial curve $\Gamma(t = 0)$ periodic in the horizontal x-direction, we amend the level set formulation to periodic boundary conditions in the level function ϕ. In this case, it is most natural to terminate the evolving field lines in a straight line square upper grid. Here, we use the same extrapolation procedure as discussed above.

3.1.3. *Interior grids.* Essentially the same construction as above can be used to generate interior grids, with suitable choice

of speed function. However, in this case our evolving front field line construction will yield interior polar grids with a center point inside the body. This is a consequence of the view of the grids as an evolving front. In order to construct such a grid, we exploit the work of Gage and of Grayson [16, 17, 19] which shows that any closed curve in the plane moving with speed equal to its curvature asymptotically becomes circular and collapses to a point. In the case of a convex initial shape, the front must always remain convex, and hence the speed function

$$F(K) = -K \tag{21}$$

is effective. In Fig. 6 we show the collapse of an ellipse to a point, and draw the field lines of the grid. In the case of a non-convex initial body, we once again have the problem of points with high curvature (in this case, high negative curvature), moving against the flow of the grid. Our remedy is to again include a threshold value which ensures that the front always moves inwards, i.e.,

$$F(K) = \min(-K, F_{\text{threshold}}). \tag{22}$$

As above, the choice of threshold value is obtained from the initial nodal displacement and is discussed below.

In general it is not true that all initial fronts collapse smoothly to a point under the speed law given in Eq. (22), since one can easily construct a dumbbell with a very narrow neck which must pinch off before the balls can collapse. However, in most examples where an interior grid is desired, the above speed function will yield smooth flow towards a single point. Examples are given in the Results section below.

The approach taken in this paper thus yields interior grids with a non-logical rectangular center point. In terms of performing calculations on the obtained grid, this forces the treatment of this point as a special case. However, since the transverse grid lines at this special point will be the same as a simple uniform polar grid, such special casing is relatively straightforward.

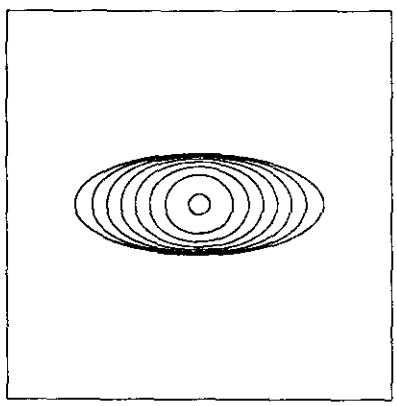

FIG. 6. Ellipse collapsing under curvature.

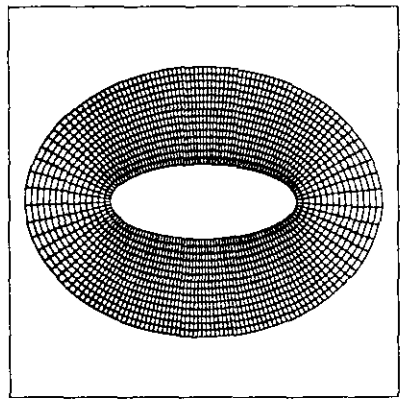

FIG. 7. Sine wave with transverse line construction.

3.2. Generating Transverse Lines

Given the above set of body-fitted field lines, we must construct the transverse lines. The goal is to balance the competing desire of orthogonality and equal area/equal size in the construction of the field lines. We stress that our construction of transverse lines produces a grid which one may later choose to adjust through a variety of elliptic and parabolic smoothers available by using more traditional techniques.

3.2.1. *Advection of nodes along gradients.* Our basic technique is to construct field lines by following trajectories along the gradient of the level function ϕ. We begin with a placement of nodes on the initial body; placement of these nodes is discussed in the next section. Let N nodes be placed at initial points (x_i, y_i), $i = 1, N$, on the body. Simultaneously with the update of the level function ϕ, we solve the N unlinked ordinary differential equations

$$\frac{dX_i}{dt} = F(K)\frac{\nabla\phi}{|\nabla\phi|} \tag{23}$$

using a second-order (Heun's) method. In Fig. 7, we show transverse lines created for an exterior grid around an ellipse using this technique. The transverse lines are, by construction, orthogonal to the body-fit field lines.

3.2.2. *Adjusting transverse lines.* For any choice of ε greater than zero, the transverse lines cannot intersect. However, for non-convex initial curves, the constructed transverse lines initially come together. The larger the choice of the smoothing parameter ε, the more the transverse gradient lines are kept apart; however, this is at a cost of large separation of the body-fit field lines. In Fig. 8, we show exterior grids generated above the periodic cosine front for various choices of the smoothing parameter. For ε small (Fig. 8a), the transverse lines

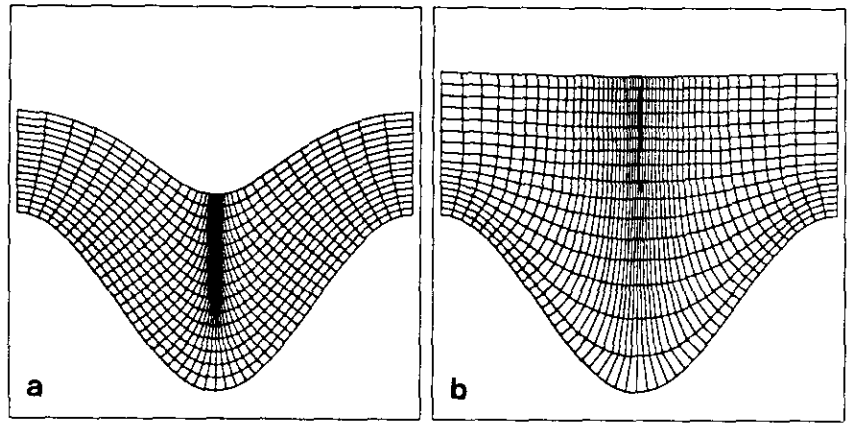

FIG. 8. Construction of transversals for propagating sine curve.

came close together. For ε large (Fig. 8b), transverse lines are kept apart, but the rapid motion of the evolving front leaves cells with large aspect ratios. Judicious choice of the smoothing parameter based on the initial node placement can solve some of these problems. However, in the case of significant curvature variation in the initial body, some transversal adjustment is desirable. This is accomplished in two ways.

In the first technique, we begin by constructing transverse lines according to the trajectory equations given in Eq. (23). Then these nodes are readjusted so that their separation distance is proportional to the separation distance along the previous field line. Thus, adjacent transverse lines that try to come together contain a penalty factor that separate them; conversely, adjacent transverse lines that separate contain a negative penalty factor that pulls them together. Penalty factors are constructed so that the net change is zero. Full use of this technique results in uniform-sized cells; however, gridline orthogonality is not maintained.

The amount of this readjustment is a user-prescribed linear weighting between the basic gradient trajectory advection and the above readjustment technique. The most desirable approach is to begin evolution away from the body with full gradient trajectory motion, and then, over several field lines, weight towards the readjustment technique.

In the second technique, we perform a readjustment according to the length of the given field line; the total length of the field line is computed, and transverse nodes are readjusted to maintain equal length spacing. Points where the derivative of the curvature vanishes, which may be calculated from a suitable difference expression acting on ϕ, act as stable points in this process and are not adjusted. Once again, this technique is linear-weighted against the basic trajectory advection. In addition, we also allow for the application of a one-dimensional diffusion operator to the field lines to separate transverse lines.

In Fig. 9, we show the use of these techniques applied to the construction of a full set of grid lines above the cosine curve.

3.3. Initial Node Placement, Time Step, and Smoothing Parameter

3.3.1. *Smooth initial boundaries.* We assume that the boundary of the body is given as a set of points on a curve, parameterized in a counterclockwise direction by $(\alpha(s), \beta(s))$,

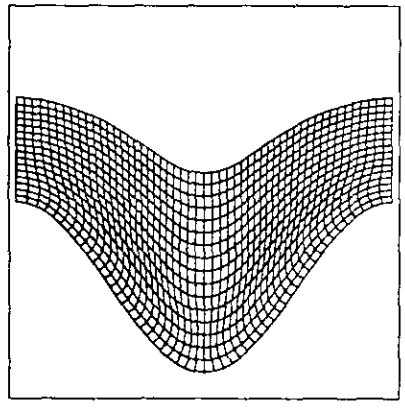

FIG. 9. Sine wave with adjusted transversals.

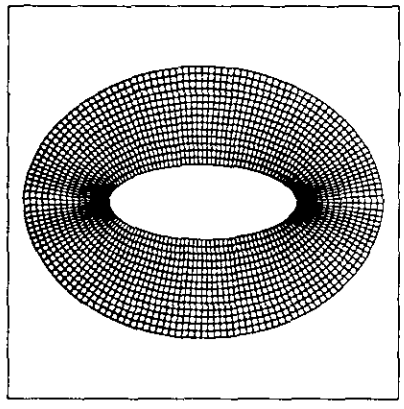

FIG. 10. Exterior grid for ellipse with nodes placed by curvature.

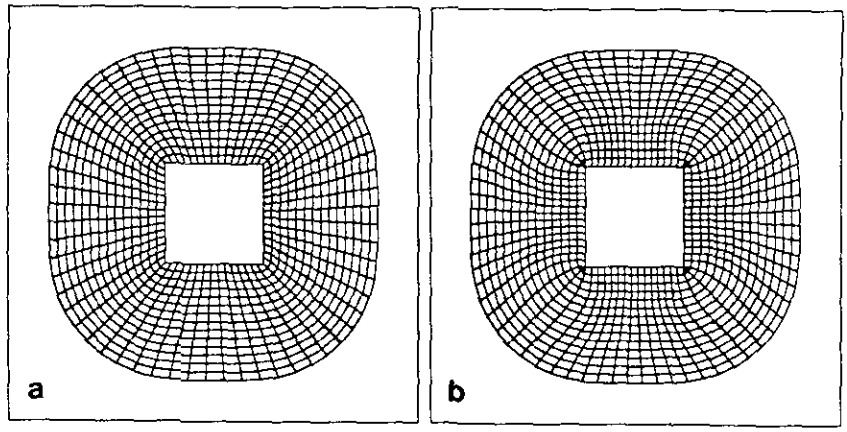

FIG. 11. Exterior grid around square.

$s \in [0, 1]$. Nodes can be placed either uniformly in the parameterization s, uniformly in arclength, or distributed according to the curvature.

In the case of node distribution according to curvature, we proceed as follows. We begin by computing the time required to traverse the curve with speed $1 - \varepsilon_{\text{node}}K$, where $\varepsilon_{\text{node}}$ is computed below. Given a requirement of N nodes on the front, we then break the total traversal time up into $N - 1$ equal time increments t_{inc}, and then retraverse the curve with the same speed, placing nodes every t_{inc} increment. The value of $\varepsilon_{\text{node}}$ is chosen by first computing the maximum and minimum values of the curvature along the initial curve, computed by central differences in the curve parameterization. A user-supplied value for the maximum ratio $\text{ratio}_{\max}$ between the largest and smallest distance between adjacent nodes is then used to compute ε through the expression

$$\varepsilon_{\text{node}} = \frac{\log \text{ratio}_{\max}}{K_{\max} - K_{\min}}, \tag{24}$$

where $K_{\max}$ and $K_{\min}$ are the maximum and minimum values of the curvature on the initial curve.

Values for both the time interval for computing body-fit field lines and ε in the speed function may now be computed on the basis of node spacing on the initial boundary. Imagine a boundary evolving with speed as above, namely

$$F(K) = \max(1 - \varepsilon K, F_{\text{threshold}}). \tag{25}$$

Let ΔT be the time elapsed between construction of the body-fit field lines. We choose ε so that the maximum variation in the speed $F(K)$ is no more than a user-specified amount; typically a value of $\varepsilon = 0.25$ works well in a large range of cases. We then choose ΔT so that average length of a transverse line leaving the initial boundary is the same as the adjacent node distance. In Fig. 10, we show an exterior grid to an ellipse with nodal placement dependent on curvature.

3.3.2. *Non-smooth initial boundaries, corners, and cusps.* In certain cases, one would like to pay special attention to corners and cusps in the initial boundaries. In this case, we alter the node placement and construction. It is desirable that the detection and treatment of cusps be automatic, which is accomplished as follows. First we traverse the initial boundary, and locate points where the curvature and normal undergo discontinuous changes. Such points are then labelled as concave or convex corners, determined by computing the sign of the cross product of the tangential vectors on either side of the point. We then make sure to place a node at such a corner, and carefully compute the normal to the boundary at this point using the construction given in [32].

Given these corner/cusps points, they can be treated as normal node points, emitting a single transverse line. However, in the case of sharp cusps, such as in airfoils, further refinement of such points may be desired. In the case of convex points for exterior grids (and concave points on interior grids), we treat the corner/cusp node point as a special polar point, and extend enough equi-spaced-in-angle transverse rays to build a fan construction from the left state to the right state. Here, the left (right) state is the limiting normal as we approach the point along the curve from the left(right); a rarefaction fan connecting these two states is constructed following the analysis and techniques for propagating fronts described in [28]. It results in a logically rectangular, body-fitted grid with special polar points at automatically detected cusps and corners.

In Fig. 11, we construct a logically rectangular, body-fitted exterior grid to a square. The corners are detected automatically; in Fig. 11a, the corners emit one transverse line each; in Fig. 11b we build the rarefaction fan. Note that in neither case will this level set approach produce a straightforward cartesian grid; this is a consequence of constructing the grid by flowing fronts.

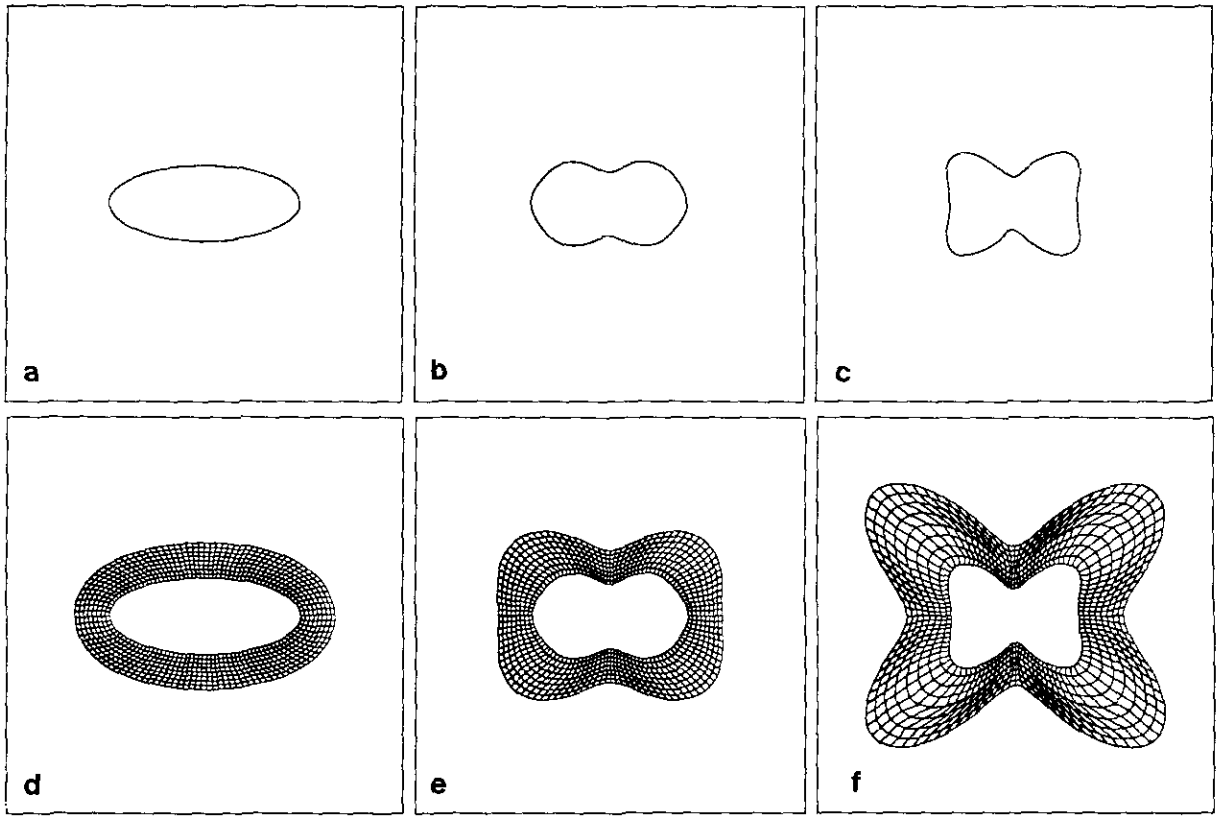

FIG. 12. Exterior moving grid: (a) $T = 0.0$, (b) $T = 1.0$, (c) $T = 2.0$, (d) $T = 0.0$, (e) $T = 1.0$, (f) $T = 2.0$.

4. EXTENSIONS TO MOVING GRIDS, REFINEMENT, AND THREE-DIMENSIONAL GRIDS

4.1. *Moving Body-Fitted Interface Grids*

A variation on the above technique can be designed to produce moving grids body-fitted to propagating boundaries and interfaces. For example, in tracking fluid boundaries, interface regions, membranes separating different physics, etc., one might wish to produce a interface-fit coordinate system such that construction of boundary conditions across the interface becomes relatively straightforward. For the sake of exposition, we imagine that the goal is to produce an exterior grid around an oscillating body, and imagine that the boundary moves under the velocity field $u(x, t)$. Here, the velocity field comes from the underlying physics, and may correspond to the flame velocity of the interface (see [28]), the solid-crystal boundary in dendritic solidification [32], a combustion front [27, 34], or a fluid interface [25, 5].

Typically, one might proceed as follows. First, construct a body-fit grid around the initial position of the boundary. Then, as the boundary moves, move all the nodes of the grid. However, for non-convex grids, this in essence becomes the method of characteristics for moving the nodes, replete with all the problems discussed in [28]. Instead, one would like to make use of the entropy condition and hyperbolic conservation law perspective outlined above.

Our technique is as follows. Given the initial front, we first use pure front propagation without constructing transverse lines and allow the boundary to propagate outwards with speed $1 - \varepsilon K$ as describe above. During this motion, we construct the travel time function (see [22]) ψ, defined as

$$\psi(x) = \{t | \phi(x, t) = 0\}. \tag{26}$$

In other words, $\psi(x)$ gives the time t when the evolving front crosses the point x. Thus, after constructing ψ, we use this as the initial level set surface $\phi(x, t = 0)$, i.e.,

$$\phi(x, 0) = \psi(x). \tag{27}$$

We now begin the calculation as follows. We evolve the level set function ϕ, initialized with the travel time function as above, according to the equation of motion

$$\frac{d\phi}{dt} + u(x, t) \cdot \nabla\phi = 0. \tag{28}$$

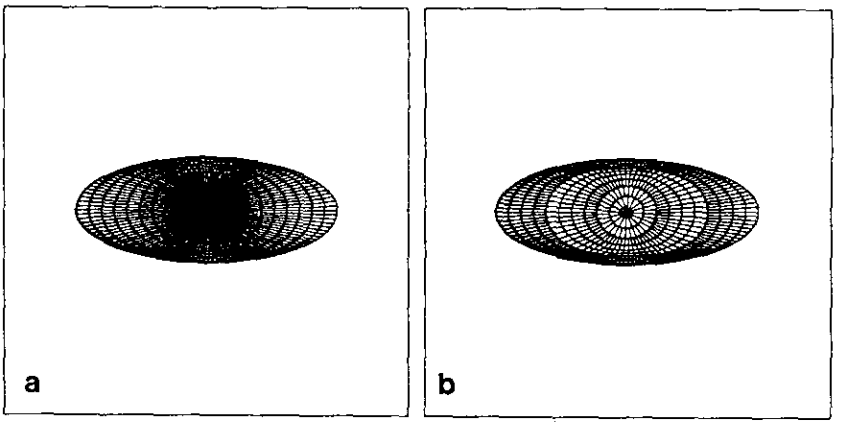

FIG. 13. Internal ellipse grid: (a) complete interior grid, (b) adjusted interior grid.

using the entropy-satisfying upwind schemes. At any time T, the position of the boundary/interface is given by the zero level set of ϕ. Furthermore, a grid aligned with the moving body/interface may be easily constructed at any time by using the mechanisms described above. Given a position of the front at time T, we can easily produce body-fit field lines by constructing contours of the level function ϕ; the values of the contour intervals are chosen to ensure equispacing in the field lines. Given these field lines, we can then rapidly produce the transverse lines using the above. As an example, in Fig. 12, we construct a body-fitted grid around a front which is oscillating under a changing electric field. In Figs. 12a–12c, we show the position of the interface as it moves. In Figs. 12d–12f, we show parts of the corresponding exterior grids.

4.2. *Refinement Grids*

In some cases, particularly with interior grids, some sort of refinement or coarsening is required. In the generation of interior grids, the focusing of all the transverse lines causes small grid cells. If the requirement of a logically rectangular grid is relaxed to allow removal of some cells, then a more desirable grid may be achieved. In Fig. 13a, we show an interior grid to an ellipse. Once the initial boundary has collapsed to a circle shape of a small enough radius, we close off the grid with a polar grid. In Fig. 13b, we remove cells below a certain size towards the center to produce a reasonable grid.

4.3. *Three-Dimensional Grids*

One advantage to this approach towards grid generation is that much of the above extends in a very natural and straightforward manner to the generation of three-dimensional grids around surfaces. Typically, one begins with a surface, and wishes to construct a grid around the surface. We follow the same technique and let the boundary be the zero level set of some function $\phi(x\ 0)$: $R^3 \rightarrow R$. We then evolve that surface according to the level set partial differential equation. The zero level surface provides the position of the front; we then construct transverse lines to this contour by following the gradient of ϕ away from the initial. If the initial surface is topologi-

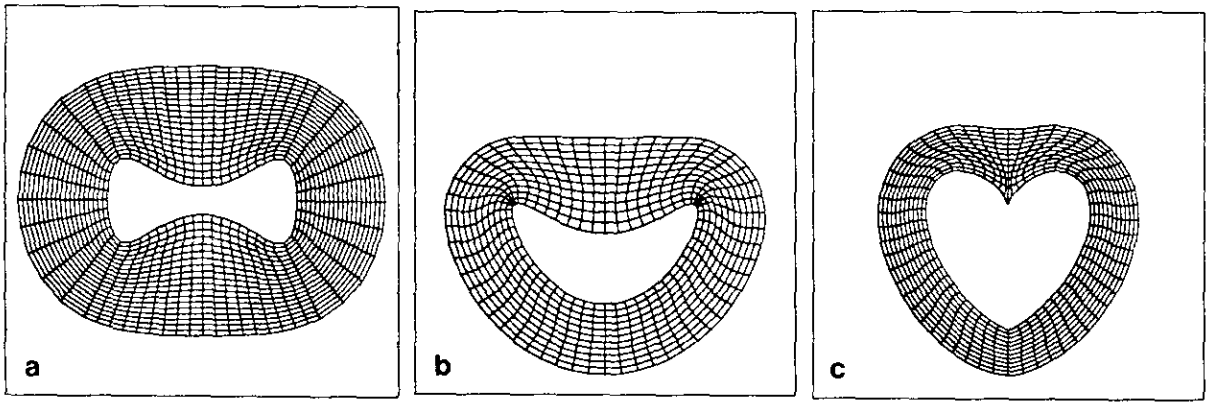

FIG. 14. External grids.

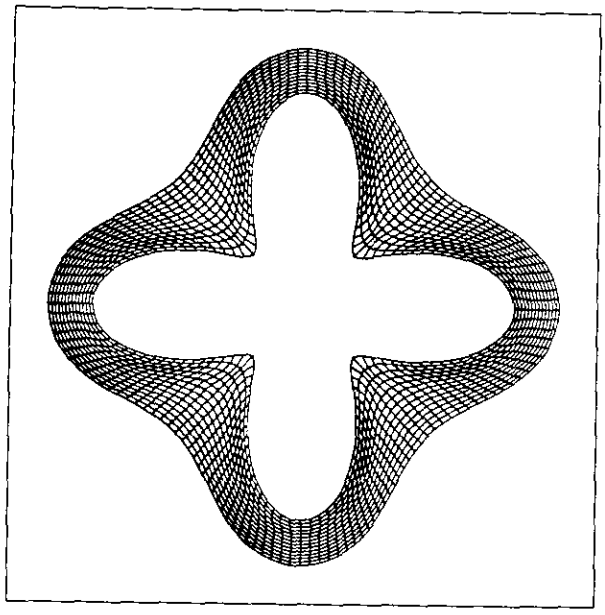

FIG. 15. External grid around star.

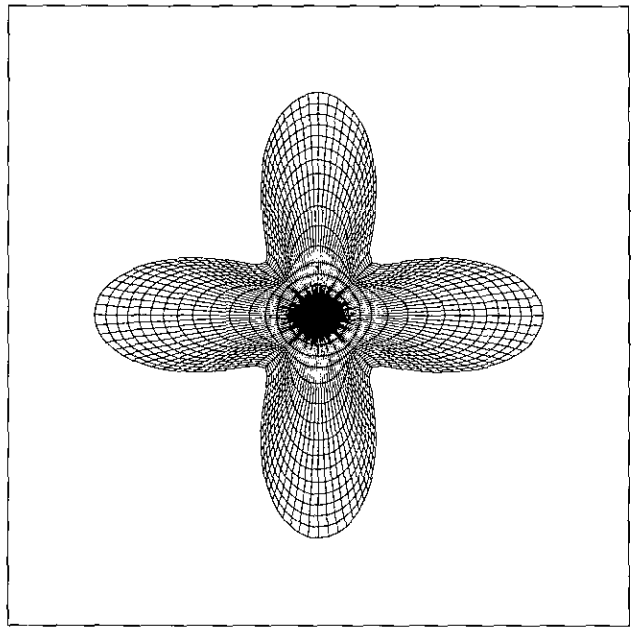

FIG. 17. Internal grid inside star.

cally equivalent to a sphere, we allow polar points in the initial node placement.

In the case of exterior grids, and exterior grids with rectangular parallel-piped far-field states, one can use a speed function of the form $1 - \varepsilon K$, where the mean curvature is used for the curvature. Once again, for non-convex initial surface, a threshold function is required, similar to the one in two space dimensions, to ensure that no part of the initial front moves inwards.

For interior grids, it is not true that a surface moving inwards under its curvature tends towards a sphere which collapses smoothly to a point, see [20, 21, 29]. We once again employ a threshold as in the two dimensional case to insure that the front always moves inwards. In the case of significantly oscillating

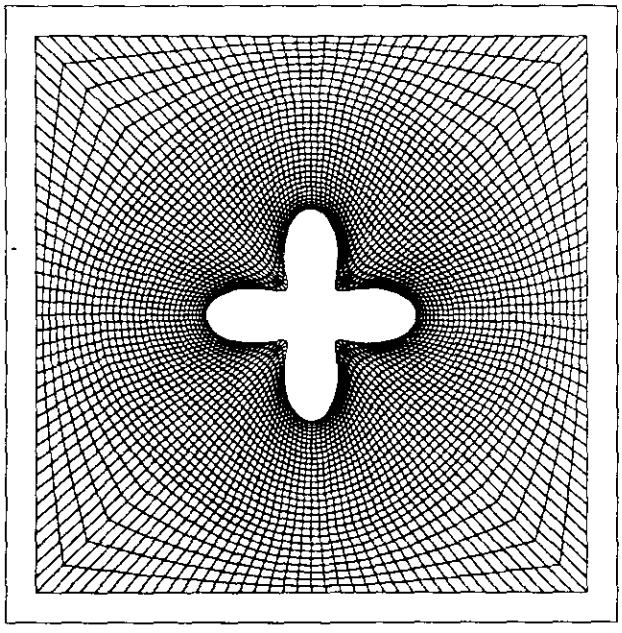

FIG. 16. External grid around star.

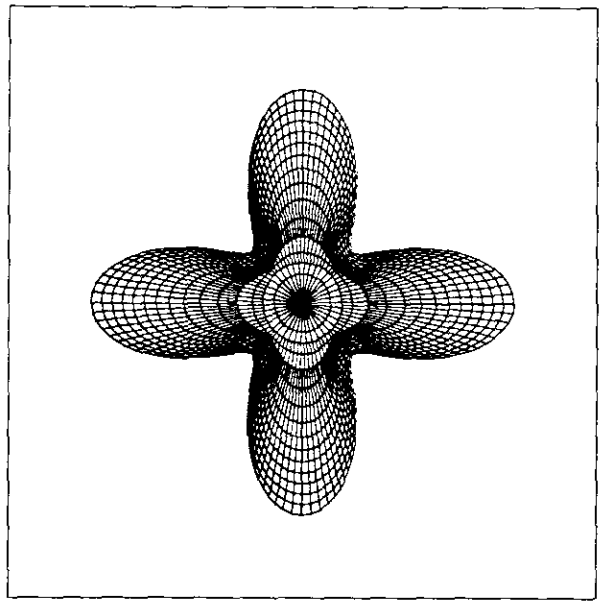

FIG. 18. Adjusted internal grid inside star.

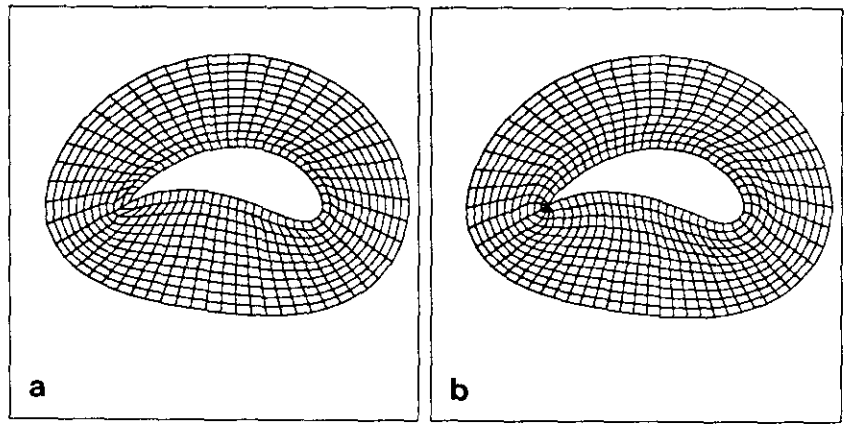

FIG. 19. External grid past single airfoil.

surfaces, the front may pinch off before the front collapses. In this case, segmenting the enclosed volume into two separate initial problems must be performed. Details of the generation of three dimensional grids using this level set approach are presented in [31].

5. RESULTS

5.1. *Examples*

In this section, we use the above algorithm to generate body-fitted grids around different bodies. In Fig. 14a, we generate an exterior grid around a depressed ellipse; in Fig. 14b, around a crescent-shaped object; and in Fig. 14c, around a cardiod.

In the next set of figures, we generate grids with respect to a four-pointed star. We begin in Fig. 15 with an external grid.

Next, in Fig. 16, we generate an external grid around the star culminating in a square.

In Fig. 17 we generate an internal grid using the four-pointed star as the outer boundary.

Finally, in Fig. 18, we remove small cells from the interior grid.

Next, we generate two types of exterior grids around a single airfoil. In Fig. 19a, an exterior grid is generated with only a single line emanating from the cusp. In Fig. 19b, we generate a full rarefaction fan at the cusp.

Next, we generate grids around shapes with corners. In Fig. 20a, we generate an exterior grid around an L-shaped region, and in Fig. 20b we generate a grid above a square wave culminating in a flat line.

In Fig. 21, we generate an exterior grid around the letter "w," showing the ability to handle inwards and outward corners. In Fig. 21a, we show the full grid; in Fig. 21b, we show a blow-up around the body.

In Fig. 22, we generate an external grid above a square cavity.

In Fig. 23, we generate an external grid around an oddly shaped object which contains inwards corners, protrusions, curved regions, and extended limbs, culminating in a square grid.

Finally, in Fig. 24, we generate a three-dimensional grid

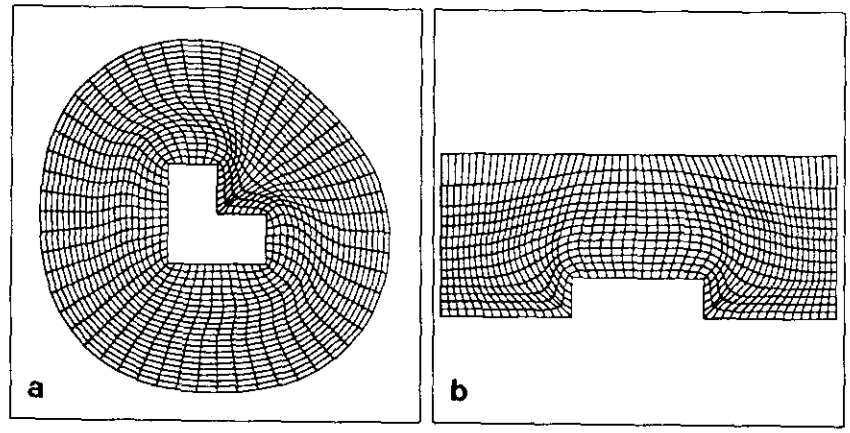

FIG. 20. External grids around corners.

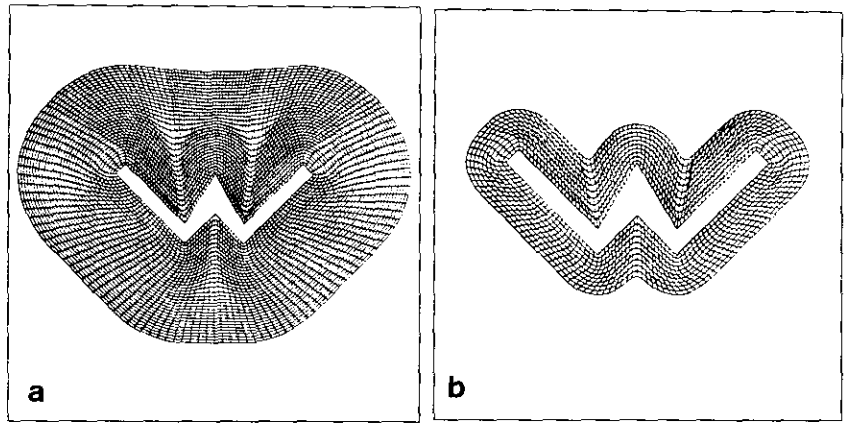

FIG. 21. External grid around letter "W."

around a non-convex body, using a speed function which depends on the mean curvature. We construct an external grid around a dumbbell; the body surface grid is shown, together with a cross-sectional slice of the three-dimensional volume grid. The viewing angle is slightly askew to give perspective.

5.2. *Timings and Numerical Parameters*

All of the above graphs were generated on grids with less than 190 points per side, with the exception of the notched external grid in Fig. 23, which required 290 points per side. No calculation required more than 1 min of cpu time on a Cray YMP. All grids were generated automatically with the same choice of numerical parameters for the smoothing of transverse lines.

6. FURTHER WORK

There are several areas where the work presented here can be significantly improved. To begin, use of more standard techniques for modifying the grid, once the basic design is achieved, might prove fruitful. In particular, the grids generated using this approach may make excellent initial grids for variational and algebraic methods. The extension of the work to three-dimensional grids requires techniques to adequately produce surface grids from which to extend transverse lines. The applica-

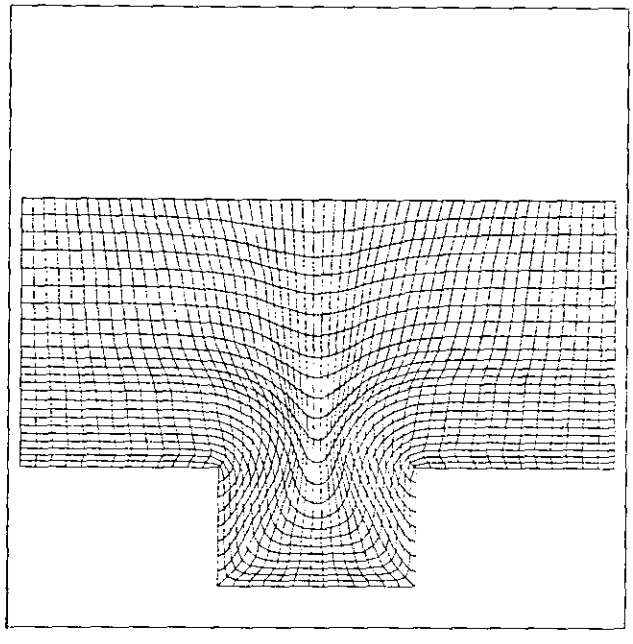

FIG. 22. External grid above square cavity.

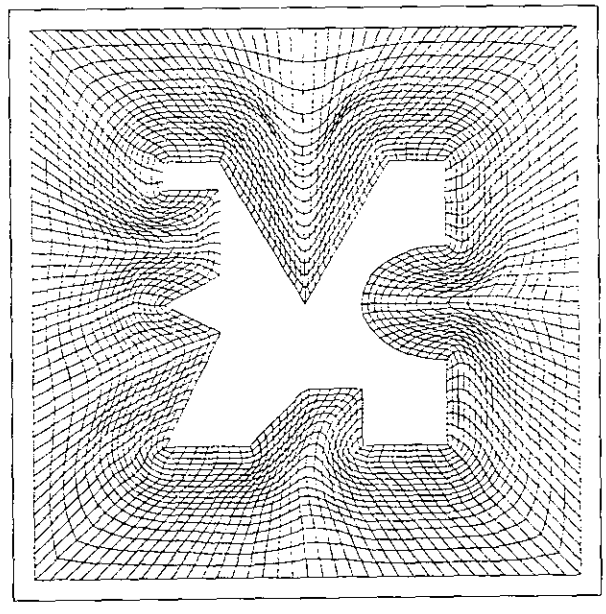

FIG. 23. External grid around notched region.

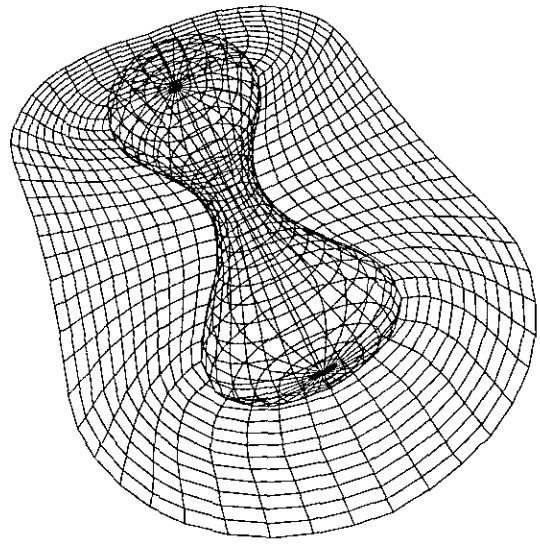

FIG. 24. *Three-dimensional grid around dumbbell.*

tion of the techniques presented here to internal grids, such as grids between two closed curves or surfaces (as opposed to interior grids, which grid the full interior of a closed curve or surface), is not straightforward, because of the essential hyperbolic advancement nature of the algorithm. Extension of this work to multiple bodies will require hybrid techniques in which single grids are patched together or allowed to overlap, similar to those commonly used in other schemes. Another possibility is the use of our techniques for grids near bodies, coupled with a transition to cartesian grids away from the body; this may be of particular use in fluid dynamics calculations where body-fitted coordinates are attractive to resolve boundary layer calculations, while far-field cartesian grids are appropriate to connect complex geometries.

The details of applying our level set approach to these ideas, as well as blending our approach with more traditional techniques to produce robust grid generation techniques, is presented in [31].

ACKNOWLEDGMENTS

All calculations were performed at the University of California at Berkeley and the Lawrence Berkeley Laboratory. We thank T. Baker, D. Chopp, O. Hald, D. Pryor, and T. Ligocki for helpful discussions.

REFERENCES

1. D. Adalsteinsson, and J. A. Sethian, A fast level set method for propagating interfaces, *J. Comput. Phys.*, submitted for publication, 1994.
2. S. Altshuler, S. B. Angenent, and Y. Giga, Mean curvature flow through singularities for surfaces of rotation, preprint, 1993.
3. B. Merriman, J. Bence, and S. J. Osher, Motion of multiple junctions: A level set approach, *J. Comput. Phys.*, to appear, 1994.
4. J. E. Castillo, *Mathematical Aspects of Grid Generation*, Frontiers in Applied Mathematics, Vol. 8, SIAM, Philadelphia, PA, 1991.
5. Y. C. Chang, T. Y. Hou, B. Merriman, and S. J. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, *J. Comput. Phys.*, submitted for publication, 1994.
6. Y. Chen, Y. Giga, and S. Goto, Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations, *J. Diff. Geom.*, **33,** 749 (1991).
7. D. L. Chopp, Computing minimal surfaces via level set curvature flow, *J. Comput. Phys.* **106,** 77 (1993).
8. D. L. Chopp, Computation of self-similar solutions for mean curvature flow, *J. Exper. Math.*, to appear, 1994.
9. D. L. Chopp and J. A. Sethian, Flow under curvature: Singularity formation, minimal surfaces, and geodesics, *J. Exp. Math.* **2** 235 (1994).
10. P. R. Eiseman, Grid generation for fluid mechanics computations, *Ann. Rev. Fluid Mech.* **17,** 487 (1985).
11. B. Engquist, and S. J. Osher, Stable and entropy satisfying approximations for transonic flow calculations, *Math. Comput.* **34,** 45 (1980).
12. L. C. Evans, H. M. Sonar, and P. E. Souganidis, Phase transitions and generalized motion by mean curvature, *Comm. Pure Appl. Math.* **45,** 1097 (1992).
13. L. C. Evans and J. Spruck, Motion of level sets by mean curvature, I, *J. Differential Geom.* **33,** 635 (1991).
14. L. C. Evans and J. Spruck, Motion of level sets by mean curvature, II, Trans. Amer. Math. Soc. **330,** 91 (1992).
15. M. Falcone, T. Giorgi, and P. Loretti, Level sets of viscosity solutions and applications, preprint, Instituto per le Applicazioni del Calcolo, Rome, 1990.
16. M. Gage, Curve shortening makes convex curves circular, *Invent. Math.* **76,** 357 (1984).
17. M. Gage and R. Hamilton, The equation shrinking convex planes curves, *J. Differential Geom.* **23,** 69 (1986).
18. Y. Giga, and S. Goto, Motion of hypersurfaces and geometric equations, *J. Math. Soc. Japan* **44,** 99 (1992).
19. M. Grayson, The heat equation shrinks embedded plane curves to round points, *J. Differential Geom.* **26,** 285 (1987).
20. M. Grayson, A short note on the evolution of surfaces via mean curvatures, *J. Differential Geom.* **58,** 555 (1989).
21. G. Huisken, Flow by mean curvature of convex surfaces into spheres, *J. Differential Geom.*, **20,** 237 (1984).
22. R. Kimmel, and A. Bruckstein, *Shape from Shading via Level Sets, Center for Intelligent Systems Report 9209, Technion—Israel Institute of Technology, June 1992.*
23. P. M. Knupp and S. Steinberg, The fundamentals of grid generation, preprint, 1993.
24. R. Malladi, J. A. Sethian, and B. C. Vemuri, Shape modeling with front propagation: A level set approach, Center for Pure and Applied Mathematics Report PAM-589, Univ. of California, Berkeley, August 1993, to appear, *Proc. Artificial Machine. Intelligence.*
25. W. Mulder, S. J. Osher, J. A. Sethian, Computing interface motion in compressible gas dynamics, *J. Comput. Phys.* **100,** 209 (1992).
26. S. Osher and J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulation, *J. Comput. Phys.* **79,** 12 (1988).
27. C. Rhee, L. Talbot, and J. A. Sethian, Dynamical study of a premixed V flame, *J. Fluid Mech.*, submitted for publication, 1994.
28. J. A. Sethian, Curvature and the evolution of fronts, *Comm. in Math. Phys.* **101,** 487 (1985).

29. J. A. Sethian, Numerical algorithms for propagating interfaces: Hamilton–Jacobi equations and conservation laws, *J. Differential Geom.* **31,** 131 (1990).

30. J. A. Sethian, Numerical methods for propagating fronts, in *Variational Methods for Free Surface Interfaces*, edited by P. Concus and R. Finn (Springer-Verlag, New York, 1987).

31. J. A. Sethian, Curvature flow and entropy conditions applied to grid generation, II, to be submitted for publication.

32. J. A. Sethian and J. D. Strain, Crystal growth and dendritic solidification, *J. Comput. Phys.* **98,** 231 (1992).

33. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation, Foundations, and Applications*, North-Holland, Amsterdam, 1985.

34. J. Zhu and J. A. Sethian, Projection methods coupled to level set interface techniques, *J. Comput. Phys.* **102,** 128 (1992).