JOURNAL OF COMPUTATIONAL PHYSICS 99, 39–55 (1992)

Discretization of Free Surface Flows and Other Moving Boundary Problems

K. N. CHRISTODOULOU AND L. E. SCRIVEN

Center for Interfacial Engineering, Minnesota Supercomputer Institute, and Department of Chemical Engineering & Materials Science, University of Minnesota, Minneapolis, Minnesota 55455

Received November 4, 1988; revised January 14, 1991

A system of elliptic partial differential equations and boundary conditions has been developed for generating boundary-fitted finite element discretizations of two-dimensional free and moving boundary problems. Terms in the differential equations are scaled for dimensional homogeneity and adjustable weighting of orthogonality, smoothness, and concentration of the coordinate mesh they govern. Grid points become finite element nodes mapped isoparametrically or subparametrically from a simple or patched computational domain. Concentration terms contain control functions and parameters that influence node spacing along each coordinate independently; overall control is by patchwise parameters and functions. Successful selection of these to follow deforming flow regions is straightforward and is illustrated by analysis of steady and transient slide coating flows.

1. INTRODUCTION

The rapid acceptance of the finite element method [1] of solving partial differential equations is largely attributable to its inherent compatibility with irregular, or "unstructured," or "neighborhood" meshes, that can be used to discretize regions of complicated shape [2]. Unfortunately, moving boundaries or moving internal interfaces are difficult to handle with unstructured meshes. In the course of solving for a succession of steady or transient states, the mesh may be redesigned and reconstructed repeatedly as the region changes shape and size. Whenever this is done the solution has to be interpolated from the old onto the new mesh, a cumbersome and diffusive process [3]. These difficulties can be circumvented by employing a curvilinear coordinate system that conforms to the boundary, a technique originally developed for the finite difference method [4, 5]. More of the difficulty can often be alleviated by dividing a complicated region into subregions, or patches, each fitted with a conforming coordinate system.

Boundary conforming meshes are generated by mapping the physical domain into a geometrically simple computational domain such that each segment of the boundary of the physical domain coincides with some coordinate line (plane) in the computational domain. The interior of the domain is divided by level lines of each computational coordinate which are computed from specified values or slopes along the boundary; see Fig. 1. There are two ways to do this: algebraic mesh generation and (differential) elliptic mesh generation. Algebraic mesh generation locates the mesh points by some simple interpolation scheme [6]. The free spine method [7], which has been a workhorse of finite element analysis of viscous free surface flows, is an example of such a technique. In common with all algebraic schemes it is computationally efficient but requires a significant amount of user interaction to define workable meshes. Moreover, it can fail by excessive mesh distortion [8] and even by crossing of mesh curves of the same family.

Elliptic mesh generation locates the mesh points by solving a partial differential equation for each computational coordinate [4]. Compared to algebraic techniques, it provides inherent smoothness of the mesh, less danger of singularity, and requires minimal input from the user. The price of a better discretization is the solution of an added—usually nonlinear—boundary value problem. The

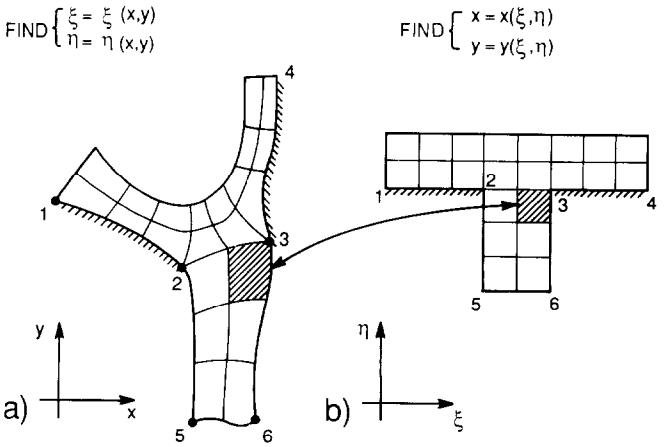

FIG. 1. Mesh generation problem: (a) in the physical region; (b) in the computational domain.

0021-9991/92 $3.00

crux of this approach is the designing of the partial differential equations for the coordinates and the boundary conditions on those equations. Relevant background is provided in Section 2, and Section 3 details the technique proposed here. The performance of this technique is demonstrated by the analysis of slide coating flows in Section 4. Implications and extensions are discussed in Section 5.

2. ELLIPTIC MESH GENERATION

The ultimate in elliptic mesh generation would be a partial differential equation system for optimizing the quality of the mesh. The goal may be to minimize the discretization error in the solution of the physical problem for a fixed number of mesh nodes. Alternatively the goal may be to minimize the number of nodes for a chosen level of discretization error. The errors of discretizing differential operators in curvilinear coordinate systems are generally formidable to evaluate. The usual approach is to employ simple error indicators tailored after the error in the first spatial derivative of a dependent variable. The most widely accepted of such error indicators quantify the smoothness, orthogonality and density of the mesh [5, 9].

The simplest of the elliptic mesh generation algorithms is conformal mapping. Conformal meshes satisfy the Cauchy–Riemann equations $\xi_x = \eta_y$, $\xi_y = -\eta_x$. They are generated by solving a pair of Laplace's equations, $\nabla^2 \xi = 0$, $\nabla^2 \eta = 0$. Conformal meshes are smooth and orthogonal and when the boundary shape is the only constraint on the mesh generated, they are usually the most efficient [4].

It is often useful to concentrate nodes in regions where the solution of the physical problem needs resolving. Conformal mapping does not allow such control of mesh spacing. Orthogonal meshes are then the simplest choice. These are less restricted than conformal meshes in two dimensions, though equally restricted in three [10]. Orthogonal meshes can be generated by stretching conformal meshes, e.g. by solving

$$\begin{aligned} \nabla \cdot S(\xi, \eta) \nabla \xi &= 0, \\ \nabla \cdot [S(\xi, \eta)]^{-1} \nabla \eta &= 0. \end{aligned} \tag{1}$$

Here $S(\xi, \eta)$ is the ratio of scale factors (or the element aspect ratio); this needs to be specified pointwise before Eqs. (1) have a unique solution [4, p. 8; 11–15]. Control of mesh spacing by distributing boundary nodes [15] is especially valuable when the mesh is divided into subregions. Nodes that lie at the corners of such subregions are often constrained, e.g., by being fixed to a feature of the physical boundary, as are points 1–6 in Fig. 1. The resulting distribution of boundary nodes must be so propagated into the interior that large deviations from orthogonality of the coordinate curves are avoided.

Godounov [16] generated orthogonal meshes by minimizing

$$\iint \left[\frac{1}{S} (x_\xi^2 + y_\xi^2 + \varepsilon S_\xi^2) + S(x_\eta^2 + y_\eta^2 + \varepsilon S_\eta^2) \right] d\xi \, d\eta.$$

This measure, without the regularization terms that involve ε, quantifies in the computational space the deviation of the mesh from an orthogonal one. As a result, it overemphasizes orthogonality in elements of large area. Godounov did not define the function $S(\xi, \eta)$ explicitly, but instead computed it during the course of the minimization. Neither Godounov's orthogonality functional nor his regularization terms are directly related to the discretization error. Moreover, they are dimensionally inhomogeneous, making it difficult to select a suitable value for ε.

When boundary segments of the physical domain are highly concave or have acute corner angles, orthogonal meshes may not be suitable because they do not allow concentrating nodes near such boundaries. Then some of the orthogonality has to be sacrificed. For example, the spacing of nodes in a mesh can also be controlled by adding to Laplace's equation suitable forcing functions $P(\xi, \eta)$ and $Q(\xi, \eta)$ to make coupled Poisson equations, $\nabla^2 \xi - P = 0$ and $\nabla^2 \eta - Q = 0$ [17–19]. A severe disadvantage of this approach is lack of control of orthogonality in the interior, particularly when the physical region is curved [20]. Visbal and Knight [21] developed a related method that delivers nearly orthogonal meshes from user-specified node distributions along certain parts of the boundary, but theirs is a two-step procedure involving successive solutions of a Poisson-like system and so it seems more cumbersome than necessary.

Brackbill and Saltzman [22] minimized a linear combination of certain measures of mesh smoothness, orthogonality, and node spacing. In two dimensions this functional is

$$\underbrace{\iint [(\nabla \xi)^2 + (\nabla \eta)^2] \, dx \, dy}_{\text{smoothness}} + \lambda_o \underbrace{\iint [\nabla \xi \cdot \nabla \eta]^2 J^3 \, dx \, dy}_{\text{orthogonality}} + \lambda_w \underbrace{\iint wJ \, dx \, dy}_{\text{node spacing}}. \tag{2}$$

Here J is the Jacobian of the mapping from the physical to the computational domain:

$$J \equiv \det \left(\frac{\partial(x, y)}{\partial(\xi, \eta)} \right) = x_\xi y_\eta - x_\eta y_\xi. \tag{3}$$

The mesh is found by solving a pair of second-order, quasi-linear partial differential equations that are the Euler equations of the minimization problem (2). The boundary

conditions that minimize (2) with respect to the edge node distribution are natural boundary conditions. If special features of the boundary or the solution there need resolving, the boundary node distribution can be controlled by adding to (2) functionals for boundary smoothness and concentration.

This method can yield impressive meshes [24]; however, it is not without difficulties. The weighting factors λ_0 and λ_w and the function $w(x, y)$ must be adjusted by trial and error to optimize performance. The minimization problem (2) may lack a solution or the solution may not be unique. Part of the trouble is that Brackbill and Saltzman's measures are dimensionally inhomogeneous, their relative contributions depending on the local area element J [25]. As a result the Euler equations of the minimization problem can lose ellipticity. A simple scaling proposed by Brackbill and Saltzman was proved to be inappropriate when disparate length scales are involved [25].

In the next section we design a new elliptic mesh generation method for free surface and moving boundary problems that combines the ideas of optimizing the mesh [24, 16] with an orthogonality measure that is directly related to discretization error, and a means of controlling the node spacing within a region and on its boundary.

3. DESIGN OF THE NEW METHOD

An additional difficulty with the Brackbill–Saltzman method lies in applying it to patched meshes, we found. When nodes at the corners of subregions have to be fixed the number of coordinate lines and hence the mean density of nodes in each subregion is set. If this density changes abruptly from one subregion to the next, the smoothness measure in (2) produces relatively large departures from orthogonality near the common corners: see Fig. 2a. When we tried to increase the orthogonality weight λ_o in (2) to correct this tendency our Newton iteration failed to converge. It was also difficult to cluster nodal points near concave boundaries. We could get solutions only by increasing simultaneously the orthogonality weight λ_o and the concentration weight λ_w. The concentration control function $w(x, y)$ was set to one everywhere to equidistribute element areas. A typical mesh is shown in Fig. 2b: all elements are of good quality, except a couple that are grossly distorted. Clearly such a mesh is unacceptable.

We believe that much of this difficulty is due to the fact that Brackbill and Saltzman optimized the mesh in the L_2-norm; i.e., they used an integral measure of the mesh quality over the entire domain. Such a procedure does not provide local control of mesh properties. What is needed is to optimize in the L_∞-norm, i.e., to minimize the distortion of the worst element in the mesh. How to optimize in the L_∞ norm is not yet well established, however. As a compromise we chose to seek L_2-functionals that are sensitive to gross

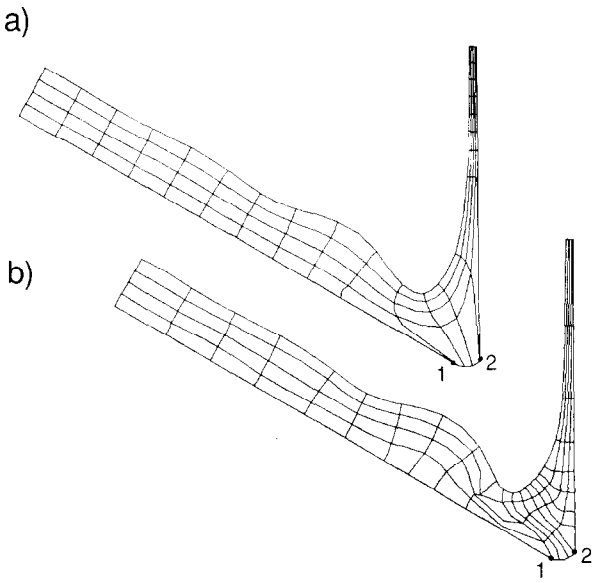

FIG. 2. Patched meshes generated by Brackbill and Saltzman's method: (a) $\lambda_0 = \lambda_w = 0$. The smoothing property of Laplace's operator causes large deviations from orthogonality at constrained points 1 and 2; (b) $\lambda_0 \neq 0$, $\lambda_w \neq 0$, $w(x, y) = 1$. Optimizing in the L_2-norm can give rise to a few grossly distorted elements.

distortions of the mesh but are comparatively insensitive to small distortions. Another source of difficulty is the inappropriateness of global scaling of the orthogonality weight λ_o [25].

3.1. *Orthogonality and Regularization*

An attractive alternative to the orthogonality measure of Brackbill and Saltzman's (in two dimensions) is

$$\iint \left[\left(\sqrt{S}\, \xi_x - \frac{1}{\sqrt{S}} \eta_y \right)^2 + \left(\sqrt{S}\, \xi_y + \frac{1}{\sqrt{S}} \eta_x \right)^2 \right] dx\, dy \tag{4}$$

which quantifies the deviation of the mesh from an orthogonal one that satisfies the generalized Cauchy–Riemann equations $S\xi_x = \eta_y$ and $S\xi_y = -\eta_x$. The Euler equations for the functions $\xi = \xi(x, y)$ and $\eta = \eta(x, y)$ that minimize this functional are the general equations for an orthogonal coordinate system (1). Hence, S here is taken as the local ratio of scale factors along the level curves of ξ and η (s is arc length):

$$S \equiv \frac{|ds/d\xi|}{|ds/d\eta|} = \sqrt{x_\xi^2 + y_\xi^2 / x_\eta^2 + y_\eta^2}. \tag{5}$$

This choice results from minimizing (4) with respect to $S(x, y)$, i.e., from the Euler equation $\partial F/\partial S = 0$, where F is

the integrand of (4). Then, as is easily established, (2) is identical to

$$\iint \frac{2}{\sin(\phi_\xi - \phi_\eta)}\, d\xi\, d\eta,$$

where $\phi_\xi - \phi_\eta$ is the angle between level curves of ξ and η in the physical region. This measure needs no scaling because it has the same dimensions as the smoothness measure adopted below (Eq. (7)). Moreover, it depends on the angle $\phi_\xi - \phi_\eta$ in the same way as does the discretization error of the first spatial derivative of a dependent variable (e.g., [5, p. 183]).

With (5) for $S(x, y)$, Eqs. (1) admit an infinite number of solutions: they are satisfied by any two sets of coordinate curves that are orthogonal [11]. In order to isolate one solution of (1) and at the same time control the concentration of nodes, small regularization terms of the form

$$-\frac{\varepsilon_1}{J}\frac{\partial}{\partial \xi}\ln[(x_\xi^2 + y_\xi^2) f(\xi)], \tag{6a}$$

$$-\frac{\varepsilon_2}{J}\frac{\partial}{\partial \eta}\ln[(x_\eta^2 + y_\eta^2) g(\eta)] \tag{6b}$$

are added to (1) (cf. [16]). Here $f(\xi)$ and $g(\eta)$ are one-dimensional concentration control functions. When $f(\xi) \equiv g(\eta) \equiv 1$, the regularization terms (6) tend to equidistribute arc-length along coordinate curves. More generally, f and g can be either specified a priori or made to depend on solution features—e.g., gradient or curvature. The minus signs in (6) are explained below. The logarithmic dependence on scale factors, an empirical modification, removes the effect of element size. The weighting factors ε_1 and ε_2 may be chosen a priori or adaptively; in (6) they are divided by J for dimensional consistency with (1) (cf. [26]). Maintaining distinct ε_1 and ε_2 allows a degree of independent control of node spacings along the two sets of coordinate curves. This is useful in situations such as film flows where the flow domain is elongated in one coordinate direction. The regularization terms (6) are not associated with the minimization of any functional; however, they can be viewed as penalty terms for the constrained optimization of orthogonality as expressed by (4).

Occasionally, it is advantageous to add to (4) a small smoothness contribution, viz.,

$$\varepsilon_s \iint [(\nabla\xi)^2 + (\nabla\eta)^2]\, dx\, dy$$

and equivalently, two regularization terms,

$$\varepsilon_s \nabla^2 \xi \quad \text{and} \quad \varepsilon_s \nabla^2 \eta \tag{7}$$

to (1). The weighting factor ε_s may be chosen a priori or adaptively. The minus signs in front of the distribution terms (6) make them compatible with the smoothness terms (7). This can be easily seen by considering one-dimensional versions of the mesh generation system (8), (9) below.

The regularization terms (6) and (7) may degrade the orthogonality that would result by minimizing (4) alone, boundary conditions permitting. Indeed, this is found to be the case when the physical region is not compatible with boundary-conforming orthogonal coordinates (e.g., when corner angles are not 90°). Fortunately, strict orthogonality is unnecessary because angles larger than 30° do not contribute significantly to the discretization error [5].

3.2. *Differential Equations and Boundary Conditions*

In summary the differential form of the mesh-generating system at which we have arrived is

$$\nabla\cdot\left(\sqrt{x_\xi^2 + y_\xi^2 / x_\eta^2 + y_\eta^2} + \varepsilon_s\right)\nabla\xi - \frac{\varepsilon_1}{J}\frac{\partial}{\partial\xi}\ln[(x_\xi^2 + y_\xi^2) f(\xi)] = 0 \tag{8}$$

$$\nabla\cdot\left(\sqrt{x_\eta^2 + y_\eta^2 / x_\xi^2 + y_\xi^2} + \varepsilon_s\right)\nabla\eta - \frac{\varepsilon_2}{J}\frac{\partial}{\partial\eta}\ln[(x_\eta^2 + y_\eta^2) g(\eta)] = 0. \tag{9}$$

Plainly in this method, a smoothness functional and a pair of concentration contraints are used to regularize an orthogonality functional. The control parameters ε_1, ε_2, and ε_s are kept small, except possibly at some boundaries where node distributions are specified. As a result the mesh produced is nearly orthogonal. We also find that specified boundary distributions of nodes are automatically propagated into the interior of the mesh with near-orthogonality maintained. There is no successive approximation of forcing functions in order to interpolate them from boundaries, as in the most flexible of previous methods. Hence Newton's method and its variants can be brought to bear on nonlinear elliptic mesh generation.

On fixed parts of the boundary the equation that defines the boundary curve (e.g., $y = y(x)$) replaces the mesh generation equation associated with the coordinate that is constant on that boundary segment. At free and unknown parts a condition from the physical problem invariably replaces the same equation. The remaining degree of freedom can be used to control the distribution of nodes along the boundary or the angle between intersecting lines at the boundary (see the next section). If large deviations from orthogonality are to be avoided, boundary node distributions ordinarily should not be specified on more than three out of the four boundary segments around each subregion [10].

If all segments of the boundary are known, the mesh generation problem decouples from the physical problem (until control parameters and functions are made adaptive). But when one or more segments are free and hence unknown, the mesh generation problem and the physical problem are coupled through the boundary conditions on those segments and must be solved simultaneously.

3.3. *Finite Element Formulation*

The finite difference method has been routinely used to solve mesh-generating systems like (8), (9). The equations are first transformed from physical coordinates (x, y) to computational coordinates (ξ, η), which can greatly complicate them (cf. [25]). In the finite element (Galerkin) method this transformation is effected by an isoparametric mapping element-by-element and the equations need be formulated only in the physical coordinates (x, y). The physical coordinates are represented in terms of finite element basis functions $\phi^l(\xi, \eta)$ that are polynomials defined on a standard square, $-1 \leqslant \xi \leqslant 1$, $-1 \leqslant \eta \leqslant 1$. The mapping (possibly time dependent) is

$$\mathbf{x} = \sum_l \mathbf{x}_l(t)\, \phi^l(\xi, \eta), \tag{10}$$

where t denotes time. The node locations $\mathbf{x}_l(t)$ are determined by requiring that the weighted residuals of Eqs. (8), (9) with respect to the same basis functions vanish:

$$\begin{aligned} R_l^\xi \equiv & \int_A \left(\sqrt{x_\xi^2 + y_\xi^2 / x_\eta^2 + y_\eta^2} + \varepsilon_s\right) \nabla\xi \cdot \nabla\phi^l \, J \, d\xi \, d\eta \\ & - \int_{\partial A} \left(\sqrt{x_\xi^2 + y_\xi^2 / x_\eta^2 + y_\eta^2} + \varepsilon_s\right) \mathbf{n} \cdot \nabla\xi \, \phi^l \frac{ds}{d\xi} \, d\xi \\ & - \varepsilon_1 \int_A f(\xi) \ln(x_\xi^2 + y_\xi^2)\, \phi_\xi^l \, d\xi \, d\eta \\ & + \varepsilon_1 \int_{\partial A} f(\xi) \ln(x_\xi^2 + y_\xi^2)\, \phi^l \, d\eta = 0 \end{aligned} \tag{11a}$$

$$\begin{aligned} R_l^\eta \equiv & \int_A \left(\sqrt{x_\eta^2 + y_\eta^2 / x_\xi^2 + y_\xi^2} + \varepsilon_s\right) \nabla\eta \cdot \nabla\phi^l \, J \, d\xi \, d\eta \\ & - \int_{\partial A} \left(\sqrt{x_\eta^2 + y_\eta^2 / x_\xi^2 + y_\xi^2} + \varepsilon_s\right) \mathbf{n} \cdot \nabla\eta \, \phi^l \frac{ds}{d\eta} \, d\eta \\ & - \varepsilon_2 \int_A g(\eta) \ln(x_\eta^2 + y_\eta^2)\, \phi_\eta^l \, d\xi \, d\eta \\ & + \varepsilon_2 \int_{\partial A} g(\eta) \ln(x_\eta^2 + y_\eta^2)\, \phi^l \, d\xi = 0. \end{aligned} \tag{11b}$$

Here A is the computational domain and ∂A is its boundary. An essential boundary condition can be imposed by replacing Eq. (11a) or (11b) at boundary nodes by the equation of vanishing (6a) or (6b), respectively. A convenient alternative is to retain the weighted residual equation but add to it a penalty term consistent with the regularization term:

$$R_b^\xi \equiv -M_1 \int_{\partial A} f(\xi) \ln(x_\xi^2 + y_\xi^2)\, \phi_\xi^l \, d\xi \tag{12a}$$

$$R_b^\eta \equiv -M_2 \int_{\partial A} g(\eta) \ln(x_\eta^2 + y_\eta^2)\, \phi_\eta^l \, d\eta. \tag{12b}$$

Here M_1 and M_2 are weighting factors much larger than ε_1 and ε_2. These two penalty terms are derived from the Galerkin forms of (6) by integration by parts. Resulting end terms can be dropped because at the ends of the ξ-interval or η-interval the mesh generation equation is replaced by either the equation of the boundary curve or a physical (e.g., kinematic) condition.

These penalty terms have an additional use that is not limited to the boundary. In isoparametric elements generated with (11) the midside nodes can get away from middles of element sides. This can give rise to uncontrolled upwinding or downwinding effects, and even to singularities in spatial derivatives of a dependent variable [27, p. 226]. Although such effects can be advantageous (as, e.g., in inertia-dominated flows, or to resolve stress singularities), they can be avoided by forcing midnodes to lie in the middle of element sides. This is accomplished by setting the concentration control function f (g) to unity and the penalty coefficient M_1 (M_2) to a large number at midside nodes (cf. subparametric mapping, Section 4.4 below).

Orthogonality at the boundary segments can be imposed naturally by dropping the first boundary integrals in (11). The second boundary integrals can also be dropped because the equations to which they belong are replaced by the equation of the boundary curve or a condition provided by the physical problem. Hence the working mesh-generating system becomes

$$\begin{aligned} R_l^\xi \equiv & \int_A \left(\sqrt{x_\xi^2 + y_\xi^2 / x_\eta^2 + y_\eta^2} + \varepsilon_s\right) \nabla\xi \cdot \nabla\phi^l \, J \, d\xi \, d\eta \\ & - \varepsilon_1 \int_A f(\xi) \ln(x_\xi^2 + y_\xi^2)\, \phi_\xi^l \, d\xi \, d\eta \\ & - M_1 \int_{\partial A} f(\xi) \ln(x_\xi^2 + y_\xi^2)\, \phi_\xi^l \, d\xi = 0 \end{aligned} \tag{13a}$$

$$\begin{aligned} R_l^\eta \equiv & \int_A \left(\sqrt{x_\eta^2 + y_\eta^2 / x_\xi^2 + y_\xi^2} + \varepsilon_s\right) \nabla\eta \cdot \nabla\phi^l \, J \, d\xi \, d\eta \\ & - \varepsilon_2 \int_A g(\eta) \ln(x_\eta^2 + y_\eta^2)\, \phi_\eta^l \, d\xi \, d\eta \\ & - M_2 \int_{\partial A} g(\eta) \ln(x_\eta^2 + y_\eta^2)\, \phi_\eta^l \, d\eta = 0. \end{aligned} \tag{13b}$$

The integrals in (13) are evaluated by Gaussian quadrature; this produces a system of nonlinear algebraic equations for the coefficients $[x_I, y_I]$ of the basis functions in (10).

4. EXAMPLE: ANALYSIS OF SLIDE COATING FLOW

Developed for the finite element discretization of viscous flows with free surfaces, the new method has been applied extensively to slide coating flow. This is an industrial precision coating operation that is routinely used in photographic manufacture. A liquid film is formed on an inclined plane, flows through a free "coating bead" bounded by upper and lower free surfaces, and is deposited on a moving sheet, displacing air as it wets the solid (see Fig. 3a).

A small vacuum—a few centimeters of water—is usually applied under the lower free surface to stabilize the bead's operation. When the lower meniscus pins at the sharp edge of the slide (Fig. 3b) the flow is usually stable. Too high a vacuum causes the lower free surface to wet the lip underside (see Fig. 3c) and may destabilize the flow to periodic disturbances in the third dimension [28].

Elsewhere we analyzed the steady, two-dimensional slide coating flow of Newtonian liquid using the older free spine method [8]. Although this method was adequate for some simple configurations, in our hands it failed when the shape of the coating bead was distorted by the pressure differential across the liquid bridge. The seat of failure was the parametrization of the lower free surface.

In contrast, the present method copes readily where free spine parametrization fails. We use it below to trace entire families of two-dimensional steady states as well as rapidly evolving transient states of slide coating flow. Its power is best illustrated by following the evolution of selected unstable steady states in an attempt to confirm the results of linear stability theory [29].

4.1. *The Physical Problem*

The Navier–Stokes system that governs the two-dimensional unsteady flow of a Newtonian liquid of density ρ, viscosity μ, and surface tension σ is (in dimensionless form)

$$-\mathrm{Re}(\partial \mathbf{u}/\partial t + \mathbf{u}\cdot\nabla\mathbf{u}) + \nabla\cdot\mathbf{T} + 3/\cos(\alpha+\beta)\,\mathbf{f} = \mathbf{0} \tag{14}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{15}$$

with boundary conditions

$$\partial K/\partial \mathbf{t} + \mathbf{u}\cdot\nabla K = 0 \qquad \text{at free surfaces} \tag{16}$$

$$\mathbf{n}\cdot\mathbf{T} = 1/\mathrm{Ca}\; d\mathbf{t}/ds + p_b\mathbf{n} \qquad \text{at free surfaces} \tag{17}$$

$$\mathbf{u} = \mathbf{u}_s \qquad \text{at solid surfaces.} \tag{18}$$

Here $\mathbf{u}$ is the velocity measured in units of $U \equiv q/h_0 = [\rho g q^2 \cos(\alpha+\beta)/3\mu]^{1/3}$, the average velocity of the fully

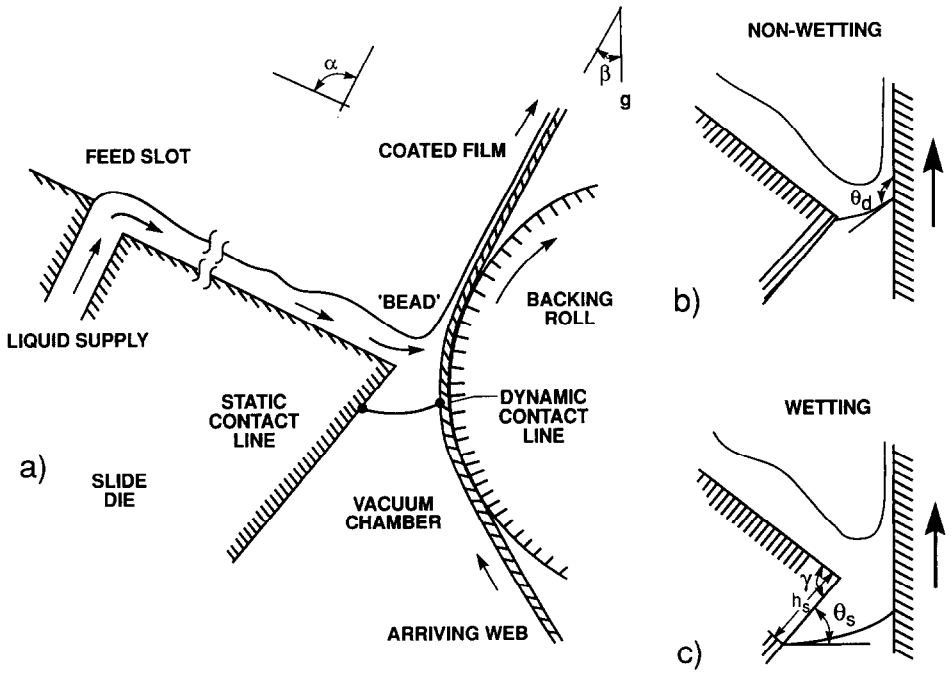

FIG. 3. (a) Slide coating flow of a single liquid layer; (b) non-wetting case; (c) wetting case.

developed film on the slide. Length is measured in units of the thickness $h_0 = [3\mu q/\rho g \cos(\alpha+\beta)]^{1/3}$ of that film; time in units of h_0/U. q is the volumetric flow-rate per unit width, β is the angle the web makes with the gravity vector, α is the angle between the slide surface and the web being coated, and g is the force of gravity per unit mass; $\mathbf{f} = -\sin\beta\mathbf{i} - \cos\beta\mathbf{j}$ is the unit vector in the direction of gravity; $\mathbf{i}$ and $\mathbf{j}$ are the unit vectors in the directions normal and parallel to the web. For an incompressibe Newtonian liquid the stress tensor is $\mathbf{T} = -p\mathbf{I} + [\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}]$, where p is the pressure and $\mathbf{I}$ the unit tensor. Both stress and pressure are measured in units of $\mu q/h_0^2$. $\mathrm{Re} = \rho q/\mu$ is the Reynolds number, $\mathrm{Ca} = \mu U/\sigma$ the capillary number, p_b the dimensionless pressure difference, and $\mathbf{u}_s$ the dimensionless velocity of solid boundaries. $\mathbf{n}$ is the unit normal to the boundary, $\mathbf{t}$ is the unit tangent, and s is the arc-length along the boundary; thus $d\mathbf{t}/ds$ measures curvature. In the kinematic condition (16), K can be defined as

$$K \equiv \eta(x, y, t) \pm 1 \tag{19}$$

because here the free surfaces have constant isoparametric coordinate $\eta = +1$ or -1 (a key connection to the mesh generation problem).

At and near the dynamic wetting line, along which liquid first contacts the moving web, we employ Navier's boundary condition:

$$\beta_{\mathrm{slip}}{}^{-1}\mathbf{t}_s \cdot (\mathbf{u} - \mathbf{u}_s) = \mathbf{t}_s\mathbf{n}_s : \mathbf{T}. \tag{20}$$

Here β_{slip} is the slip coefficient and $\mathbf{t}_s$ and $\mathbf{n}_s$ are the unit tangent and normal to the solid surface. The angle of contact between the liquid/air interface and the solid surface is also needed as a boundary condition on the free surface shape at the wetting line, the location of which is unknown (another key connection to the mesh generation problem):

$$\mathbf{n}_s \cdot \mathbf{n}_f = \cos\vartheta_d. \tag{21}$$

Here $\mathbf{n}_f$ is the outward unit normal at the wetting line to the visible free surface. Both β_{slip} and ϑ_d can be regarded as empirical parameters that have to be measured experimentally.

When the coating liquid wets the underside of the lip, as in Fig. 3c, the contact angle θ_s and the cut-back angle γ are also needed for the description of the flow. With the wetted distance h_s and the contact angle θ as shown in Fig. 3, the boundary conditions at the static contact line become

$$(\theta_s - \theta)\, h_s = 0 \qquad \text{with} \quad \theta_s - \theta \geqslant 0,\ h_s \geqslant 0. \tag{22}$$

Here $\theta_s = \cos^{-1}(\mathbf{n}_f \cdot \mathbf{n}_s)$ and the unit normals $\mathbf{n}_f$, $\mathbf{n}_s$ are evaluated at the static contact line. Equation (22), which goes back to Gibbs, is a complementarity condition between $\theta_s - \theta$ and h_s [30, 31]: when the contact line is pinned, i.e., h_s is equal to zero, θ can assume values less than θ_s; when there is wetting, h_s is positive and θ must be exactly equal to θ_s. The contact angle θ_s can also be regarded as an empirical parameter.

At an inflow boundary placed sufficiently far upstream on the slide a semiparabolic velocity profile is specified. Finally, at an outflow boundary placed sufficiently far downstream, the traction is set to zero and the asymptotic free surface slope is imposed naturally as detailed elsewhere [8].

4.2. *Finite Element Formulation*

Steady and transient states of the system (14)–(23) were computed by the Galerkin/finite element method and Newton iteration. The velocity, pressure, and position unknowns were expanded in terms of the basis sets $\phi^i(\xi, \eta)$ and $\psi^k(\xi, \eta)$:

$$\mathbf{u} = \sum_i \mathbf{u}_i(t)\, \phi^i(\xi, \eta), \tag{23a}$$

$$p = \sum_k p_k(t)\, \psi^k(\xi, \eta). \tag{23b}$$

The basis functions were nine-node biquadratics for ϕ^i and four-node bilinears for ψ^k [32]. These functions were constructed on the standard (ξ, η) square domain. This square was mapped into each of the deformed quadrilateral elements in the flow domain by the time-dependent isoparametric mapping (10).

Time derivatives $\partial/\partial t$ at a fixed location in Eqs. (14) and (16) were transformed to time derivatives at fixed isoparametric coordinates, denoted by an overdot ($\dot{\mathbf{x}}$ denotes mesh velocity):

$$\partial\mathbf{u}/\partial t = \dot{\mathbf{u}} - \dot{\mathbf{x}} \cdot \nabla\mathbf{u}. \tag{24}$$

The weak (Galerkin) form of (14), (15), and (24) was then obtained by multiplying with the appropriate basis functions, integrating over the physical domain, and applying the divergence theorem to the momentum equation:

$$\mathbf{R}_i^M \equiv \int_A \left\{ \mathbf{T} \cdot \nabla\phi^i + \left[\mathrm{Re}(\dot{\mathbf{u}} + (\mathbf{u} - \dot{\mathbf{x}}) \cdot \nabla\mathbf{u}) - \frac{3\mathbf{f}}{\cos(\alpha+\beta)} \right] \phi^i \right\} dA - \int_{\partial A} \mathbf{n} \cdot \mathbf{T}\, \phi^i \frac{ds}{d\xi}\, d\xi - \int_{\partial A} \mathbf{n} \cdot \mathbf{T}\, \phi^i \frac{ds}{d\eta}\, d\eta = \mathbf{0} \tag{25}$$

$$R_k^c \equiv \int \psi^k \nabla \cdot \mathbf{u}\, J\, d\xi\, d\eta = 0 \tag{26}$$

$$R_m^k \equiv \int_{\partial A} \mathbf{n} \cdot (\mathbf{u} - \dot{\mathbf{x}})\, \phi^m \frac{ds}{d\xi}\, d\xi = 0. \tag{27}$$

M, C, and K denote momentum, continuity, and kinematic residuals. Essential boundary conditions were imposed by replacing the corresponding weighted residual equation with the desired velocity or free-surface specification. Natural boundary conditions were imposed through the boundary integrals in (25) as described by, for example, Ruschak [33].

4.3. *Evaluation of the Basis Function Coefficients*

The weighted residuals (25)–(27) involve the unknown coefficients of the basis functions in (10) and (23). There are as many momentum equations (25) as velocity unknowns, as many continuity equations (26) as pressure unknowns, and as many kinematic equations (27) as free surface nodes. To close the system, equations are needed for the cartesian coordinates of interior nodes and for one of the two coordinates of each of the free surface nodes. These are provided by the mesh generation system (13). There are as many mesh generation equations (13) as nodal coordinate unknowns. At free surfaces the kinematic equation (27) replace equation (13b), except at contact lines where the contact angle condition (21) or (22) replaces (13b).

Biquadratic isoparametric elements were first chosen to approximate the geometry in (10). Subparametric elements for the interior elements were subsequently found to be more efficient as discussed below. Area integrals in (10), (25)–(27) were evaluated by nine-point Gaussian quadrature and line integrals by three-point quadrature. The result of integration is a system of differential and algebraic equations of index two [34] $\mathbf{f}(t, \mathbf{y}, \dot{\mathbf{y}}) = \mathbf{0}$, for the vector of time dependent coefficients $\mathbf{y} = [u_i, v_i, p_j, x_l, y_l]^T$. Steady states of the system were found by Newton iteration with zero-order continuation in parameters and Hood's [35] frontal solver. The parameter step size was adjusted manually to achieve convergence in four to six iterations. The very first initial guess was a solution calculated by the free spine method [8].

Our choice of a time-stepping method was the differential-algebraic system solver DASSL [36], which is based on a variable time-step, variable order, backward difference method. This solver requires a linear combination of the Jacobian, or sensitivity matrix $\partial \mathbf{f}/\partial \mathbf{y}$, and the mass, or the basis function overlap matrix, $\partial \mathbf{f}/\partial \dot{\mathbf{y}}$.

Consistent initial conditions and time derivatives were calculated by taking two steps with the first-order backward difference algorithm and suppressing DASSL's time truncation error control during those steps. Pressures and coordinates of nodes other than those on free surfaces were excluded from DASSL's time-truncation error estimation but not from the convergence test of the Newton iteration, as suggested by Petzold [37]. Doing so made DASSL perform faster and more smoothly, i.e., abrupt changes of the time-step size were avoided.

4.4. *Performance of the New Mesh Generation Method*

Steady states. The elliptic mesh generation system (3.10) is demonstrably more flexible than the free spine method [7] in the computation of steady states of the slide coating flow. First we consider the case where the lower meniscus remains pinned at the sharp edge of the slide die. A typical tessellation of the flow domain into finite elements generated by the spine method is shown in Fig. 4a (see Table I for parameter values). The location of the base point of each spine and its orientation and spacings along it, had to be chosen a priori or generated adaptively for each steady state as we detail elsewhere [8].

As the gap is widened, the liquid bridge bulges downwards under the combined action of gravity and applied vacuum; see Fig. 4. As the distortion grows it becomes impossible to parametrize both the upper meniscus and the lower meniscus with the same straight spines: the angles between coordinate curves at the slide edge soon become too large and Newton iteration diverges. Designing curved spines is tedious and success is not guaranteed. When the calculation fails to converge it is not clear whether the limit point is caused by the discretization or whether it is a real feature in the solution space of the flow problem.

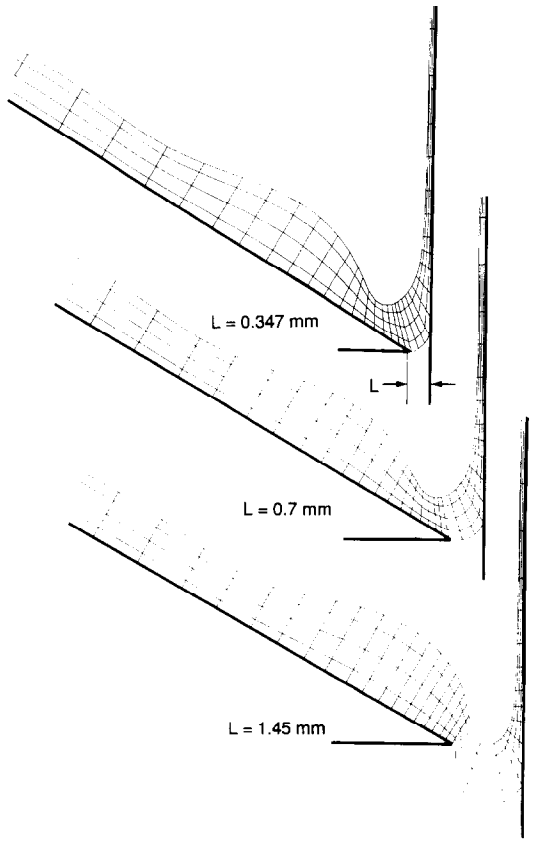

FIG. 4. Sequence of finite element meshes of slide coating flow for different gap-widths generated by the free-spine method of Kistler and Scriven [7].

TABLE I

Parameters

(a) *Typical slide coater system and the base case*

Parameter			Range	Base case
Final film thickness	μm	h	30–200	115.3
Web speed	m/s	U_w	0–5	1.67
Pressure difference	Pa	p_b	0–40	161.1
Clearance	mm	L	0.1–0.4	0.347
Viscosity	mPa.s	μ	0.5–100	8.717
Density	gr/cm^3	ρ	0.9–1.3	1.13
Surface tension	dyn/cm	σ	20–70	70.1
Dynamic contact angle	deg.	θ_d	140–160	160
Slide–web angle	deg.	α	60–80	60
Web–gravity angle	deg.	β	0–25	0

(b) *Dimensionless*

Group	Definition	Base case
Reynolds number	$\mathrm{Re} \equiv q\rho/\mu$	25
Capillary number	$\mathrm{Ca} \equiv \mu U/\sigma$	0.0247
Pressure difference	$P_b \equiv h_0^2 p_b/\mu q$	-90
Clearance	$L_g \equiv L/h_0$	0.358
Draw-down ratio	$U_R \equiv U_w/U$	8.4
Dynamic contact angle	θ_d	160
Slip-coefficient	β_{slip}	0.01
Slip length	d_{slip}	one element
Slide–web angle	α	60°
Web–gravity angle	β	0°

Units

Length	$h_0 \equiv [3\mu q/\rho g \cos(\alpha+\beta)]^{1/3}$	0.969 mm
Velocity	$U \equiv [\rho g q^2 \cos(\alpha+\beta)/3\mu]^{1/3}$	19.905 cm/s
Time	$T \equiv h_0/U$	4.869 ms

Figure 5 shows the same computation with the elliptic mesh generation system (13). The control functions used in (13) were $f = (1 + a\xi_j)/(1 + a)$ on the slide, $f = 1$ in the bead, and $f = [1 + a(1 - \xi_j)]/(1 + a)$ in the web region, with $a = 40$. Here ξ_j was the element column number in a subregion divided by the total number of element columns in that subregion. (Likewise, although not used here, η_j would be the row number divided by the total number of rows in that subregion.) These choices of f caused the nodes and thus the elements to concentrate toward the bead region in order to resolve the stress singularities at the contact lines. Increasing the parameter a caused more and more clustering of nodes toward the bead. In the cross-stream direction g was set to unity in all regions; this enforced equidistribution of arc length in the filmwise direction. The distribution prescribed by $f(\xi)$ was also enforced at the lower boundary by setting the penalty parameter M_1 to 10^3 there. M_1 and M_2 were set to zero on all other boundary segments.

As the gap widened and the bead region elongated, elements were added in the streamwise direction at the bead region; compare states in Fig. 5. Coordinate curves between elements automatically spread out to accommodate the newly added elements. No change of control functions or parameters was necessary. Refinement in both coordinate directions was also easy; Fig. 6 shows three different meshes with $(15 + 2 + 9) \times 3$, $(20 + 3 + 12) \times 4$, and $(30 + 5 + 18) \times 6$ elements. The predicted dynamic contact line locations were 0.336, 0.337, and 0.3394. The reason the location appears not to converge is that refinement changes the Navier slip length because slip was employed only in the one element that extends downstream from the dynamic contact line. Employing slip over the two elements

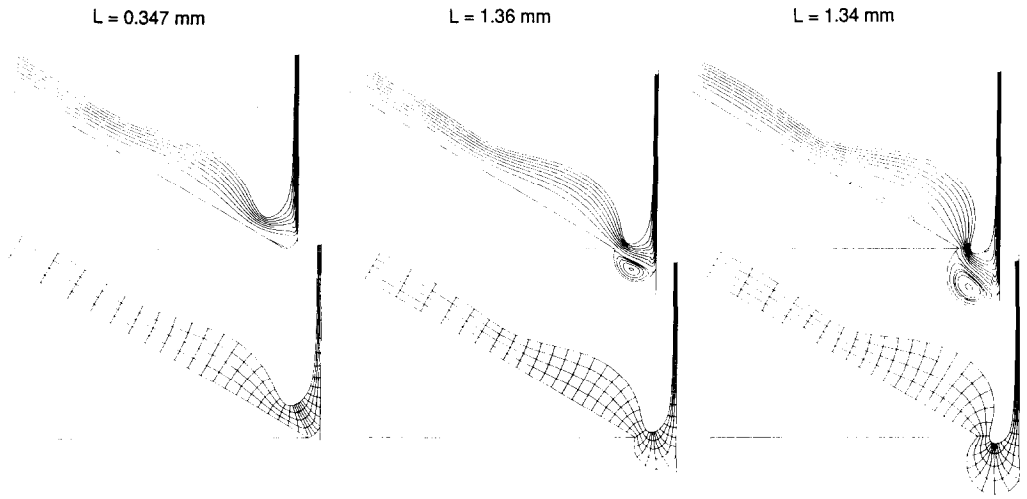

FIG. 5. Sequence of states and finite element meshes of slide coating flow along a family of states of varying gap width. The last state is past the turning point (see Fig. 7). Meshes generated by the new elliptic system with $f = (1 + a\xi_j)/(1 + a)$ on the slide, $f = 1$ in the bead, and $f = [1 + a(1 - \xi_j)]/(1 + a)$ in the web region, with $a = 40$, $\varepsilon_s = 0.05$, $\varepsilon_1 = 0.25$, and $\varepsilon_2 = 100\varepsilon_1$.

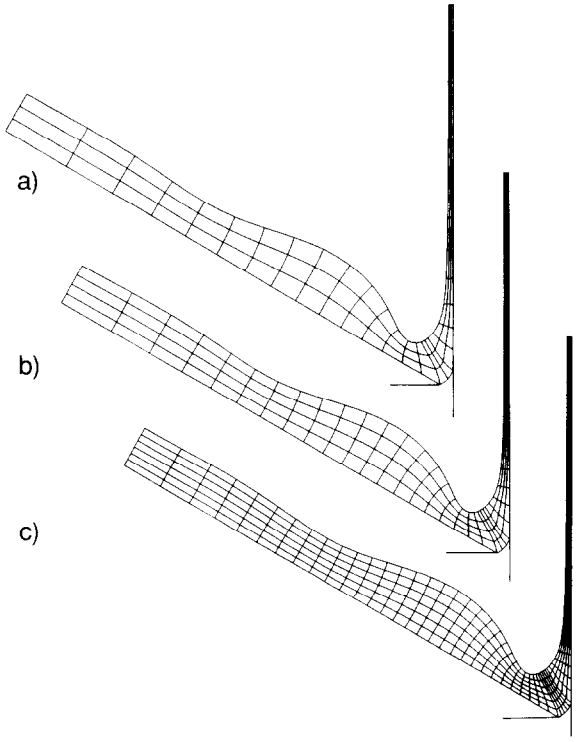

FIG. 6. Mesh refinement in both directions: (a) $(15+2+9)\times 3 = 78$ elements; (b) $(20+3+12)\times 4 = 140$ elements; (c) $(30+5+18)\times 6 = 318$ elements.

extending downstream makes the slip length of the coarsest and finest meshes approximately equal, we found, and changed the predicted dynamic contact line position to 0.3357. This is within 0.1% of the prediction from the coarsest grid. Clearly a more consistent treatment of the slip condition is desirable to achieve the optimal rate of convergence. An attractive option is to use singular elements that can represent accurately the apparent singularity at the contact line [27]. The second of the meshes just mentioned was used in most of the calculations that follow. However, the number of columns of elements in the bead region was varied as parameters that affected the length of the bead region were changed.

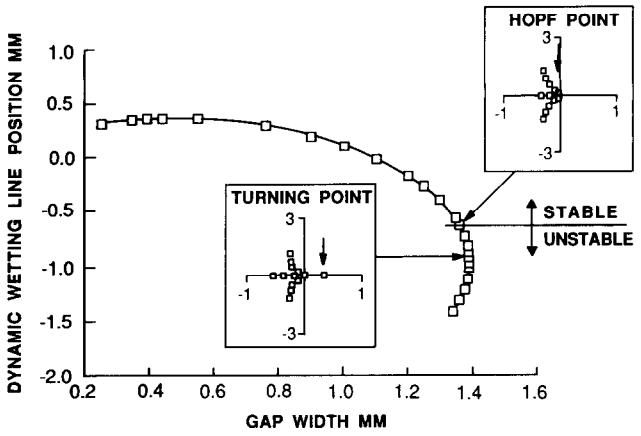

FIG. 7. Turning point along the family of states of varying gap width.

It proved possible to trace the entire family of states without changing the control functions or control parameters. Continuation revealed the existence of a limit point at which this family of solutions turns back toward lower values of the gap width; see Fig. 7. Stability and multiplicity of this family of states have been discussed elsewhere [29]. Two more examples that test the capabilities of the elliptic system are shown in Fig. 8 and 9. Control parameters and functions were identical to those of the base case.

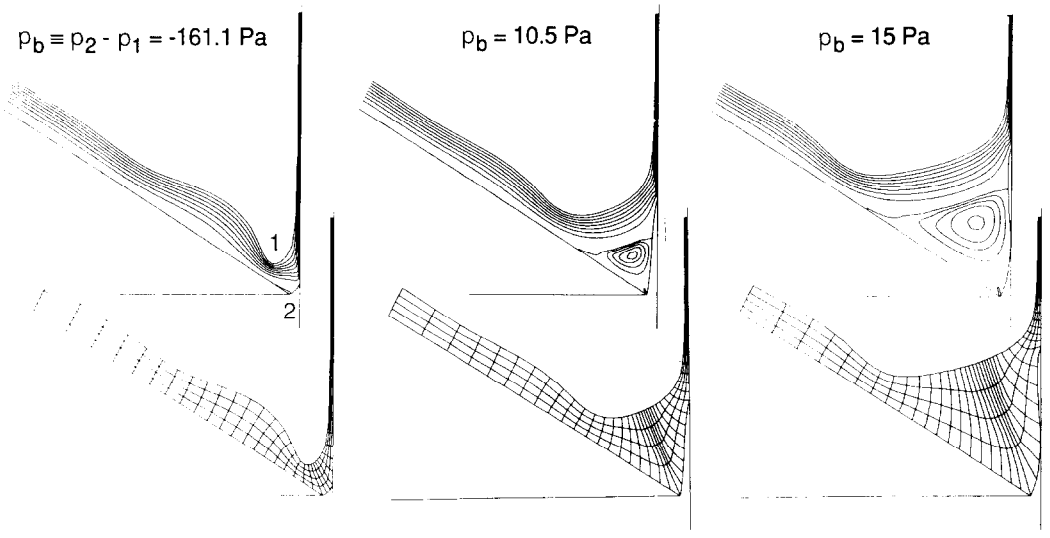

FIG. 8. Sequence of steady states and finite element meshes along a family of varying pressure difference between upper and the lower free surface.

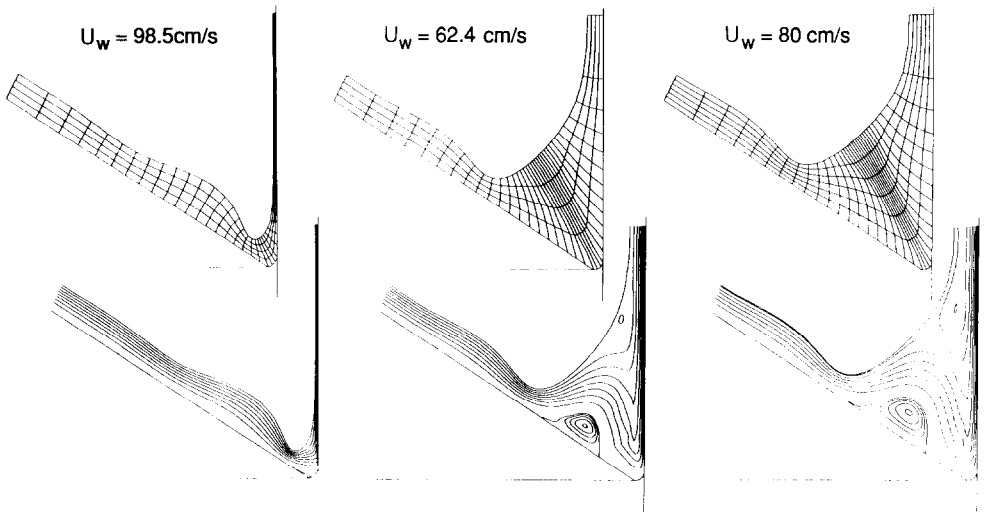

FIG. 9. Sequence of steady states and finite element meshes along a family of varying web speed. The last state is past the turning point (see Christodoulou and Scriven [29]).

Cost comparisons are revealing. Both the spine and elliptically generated discretizations have $(20+3+12)\times 4 = 140$ elements. Spine parametrization leads to 1539 unknowns; the elliptic mesh leads to 2738 (the difference being the number of independent node coordinates). Using the one-processor CRAY-2 at the University of Minnesota, the free spine calculation required 2.4 CPU seconds per Newton iteration, the elliptic mesh generation system 5 s. The 100% increase is certainly justified if the solutions that could not be found by the spine parametrization are required. The increase may also be partly justified by savings from trial-and-error adjustment of spines. However, the difference can be reduced by using subparametric mapping.

The number of unknown nodal coordinates can be significantly reduced by adopting subparametric elements in the interior of the mesh. This is done by making all element sides, that are not boundary segments, straight lines and placing the mid-nodes at the middles of the corresponding element sides, i.e.,

$$\mathbf{x}_2 = 1/2(\mathbf{x}_1 + \mathbf{x}_3), \qquad \mathbf{x}_4 = 1/2(\mathbf{x}_1 + \mathbf{x}_7),$$
$$\mathbf{x}_6 = 1/2(\mathbf{x}_3 + \mathbf{x}_9), \qquad \mathbf{x}_2 = 1/2(\mathbf{x}_1 + \mathbf{x}_3).$$

The center node of each element is positioned at

$$\mathbf{x}_5 = 1/2(\mathbf{x}_1 + \mathbf{x}_3 + \mathbf{x}_7 + \mathbf{x}_9) - 1/4(\mathbf{x}_2 + \mathbf{x}_4 + \mathbf{x}_6 + \mathbf{x}_8),$$

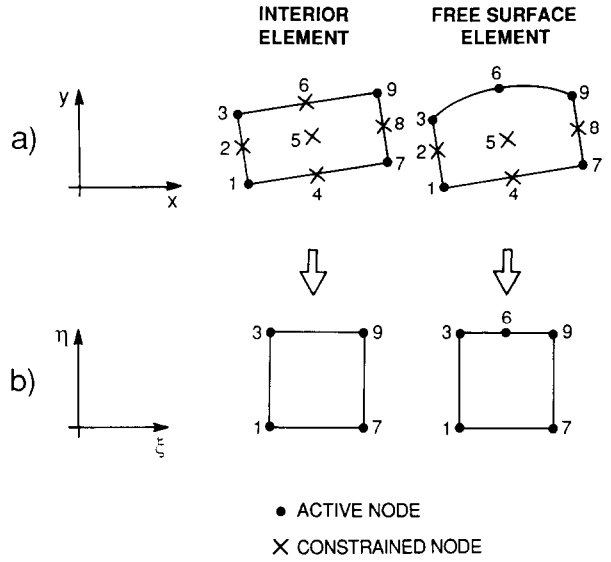

FIG. 10. Subparametric mapping.

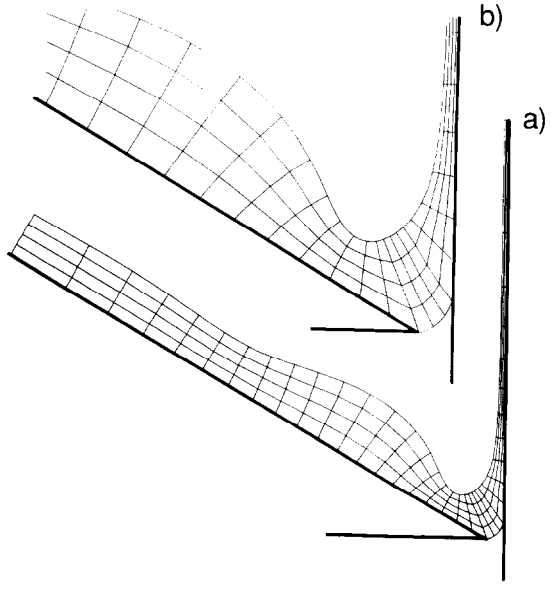

FIG. 11. (a) Elliptically generated finite element mesh with subparametric elements; (b) enlargement of the bead region.

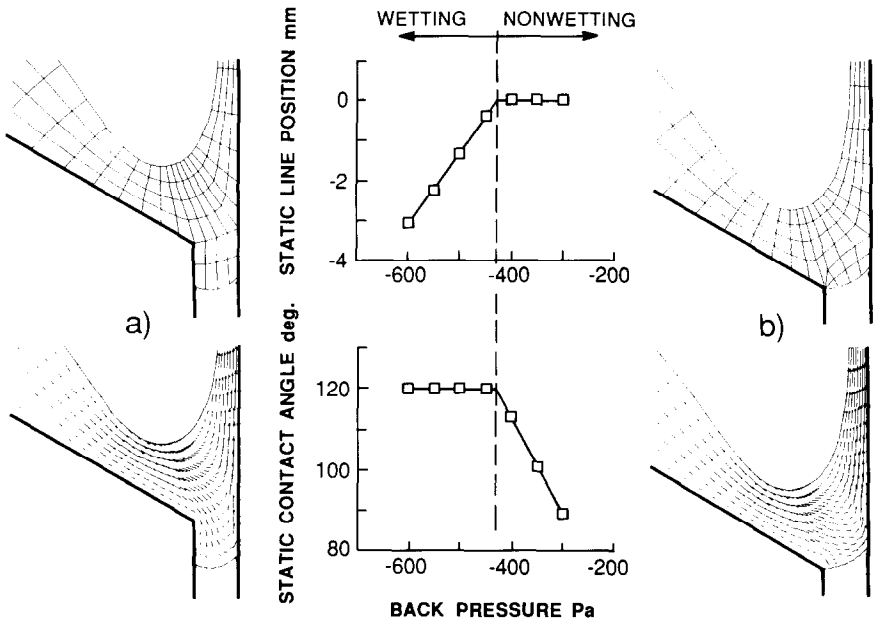

FIG. 12. Static wetting line position and static contact angle as functions of applied pressure difference. Mesh was generated by the new elliptic system with isoparametric elements and $f = [1 + \sinh(a\xi_j)]/[1 + \sinh(a)]$ on the slide, $f = 1$ in the bead, $f = [1 + \sinh(a - a\xi_j)]/[1 + \sinh(a)]$ on the web, $a = 5$, $\varepsilon_s = 0.1$, $\varepsilon_1 = 0.25$, and $\varepsilon_2 = 10\varepsilon_1$.

where $\mathbf{x}_i$ is the vector of cartesian (x_i, y_i) coordinates of local node i; see Fig. 10. This choice minimizes distortion of the element [27, p. 200], which is a major contributor to discretization error. Thus subparametric elements are not only less costly but also more accurate than isoparametric elements. Moreover, they avoid the midside node migration described in Section 3.4. Hence with them the penalty terms (12) are not needed away from the boundary.

Figure 11 shows the base case with subparametric elements. The level curves of ξ are straighter than those generated with isoparametric elements in Fig. 6, though at the expense of some loss of orthogonality at the lower boundary. The reason is that the same biquadratic weighting functions that were used in the isoparametric elements were used in the weighted residuals (13) with subparametric elements, a procedure that replaces Galerkin forms by so-called Petrov–Galerkin ones [38]. The choice of weighting functions, however, is not crucial for the mesh generation equations. Simply, when quadratics are used, orthogonality—the main measure in the mesh generating system—is weighted more at free surfaces and less at solid boundaries, causing straightening of coordinate curves where they are approximately orthogonal to a free surface. Adopting subparametric elements reduces the number of unknowns in the base case (the number of field variables being unchanged) from 2738 to 1944 and the CPU time from 5 to 2.9 s per iteration. This is a 40% improvement over the isoparametric elements and is only 16% more costly than the spine parametrization.

The system was also tested in a Y-shaped domain which arises when the lower meniscus of the slide coating bead wets the underside of the die lip. A mesh configuration capable of parametrizing both wetting and non-wetting states eliminates the cumbersome task of changing meshes and transferring data between them. Such a mesh was

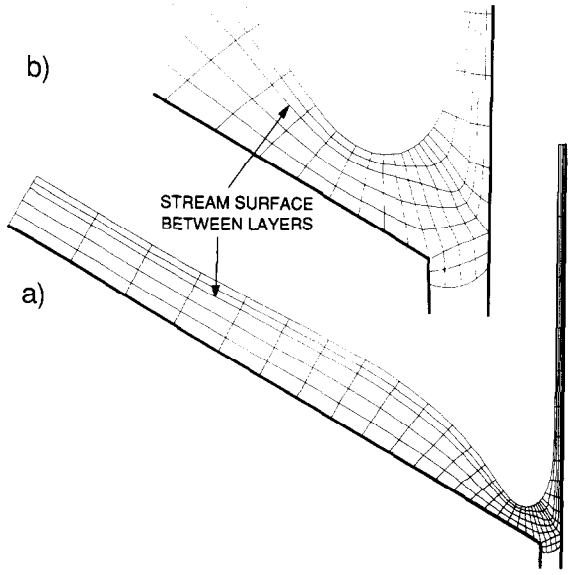

FIG. 13. Finite element mesh for two-layer slide coating flow generated by the new elliptic system with subparametric elements and $f = [1 + \sinh(a\xi_j)]/[1 + \sinh(a)]$ on the slide, $f = 1$ in the bead, $f = [1 + \sinh(a - a\xi_j)]/[1 + \sinh(a)]$ on the web, $a = 6$, $\varepsilon_s = 0.1$, $\varepsilon_1 = 0.25$, and $\varepsilon_2 = 10\varepsilon_1$.

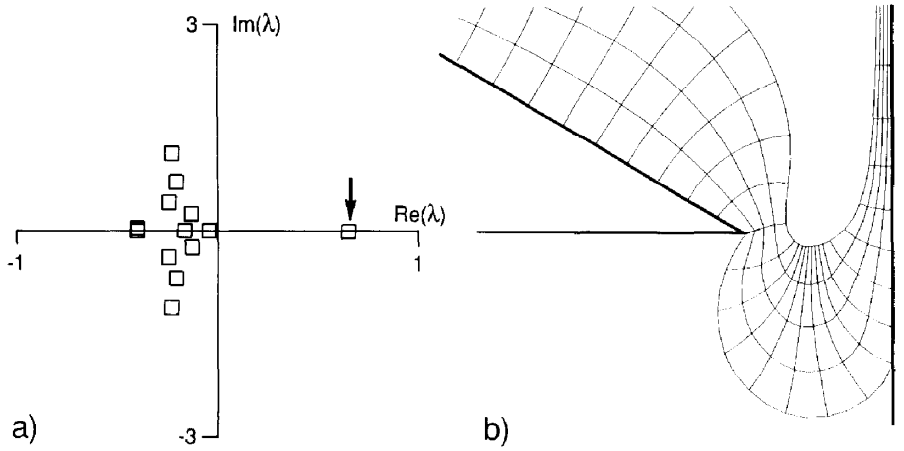

FIG. 14. Initial (unperturbed) state for transient analysis and its eigenspectrum: the state is made unstable by the positive real eigenvalue.

achieved by allowing the nodes along the lower side of the lip to collapse onto the die edge when the physics dictated that the die face was not wetted. Figure 12 shows the static contact line position and static contact angle as functions of the pressure difference applied across the liquid bridge. The complementarity condition (22) was obeyed, as intended. When the meniscus pins at the edge, coalescing nodes give rise to degenerate, or singular, elements which are advantageous for resolving the stress singularity there [27].

Figure 13 shows another application of the subparametric mapping: a flow of two superposed, miscible liquid layers, with the lower layer wetting the underside of the lip. At the interface between the two layers, Eq. (13b) was replaced by the kinematic condition (27). Between the two layers there is no interfacial tension; thus the interface can curve sharply as it approaches the wetting line. When the mesh was not sufficiently refined in this region, "wiggles" appeared in the dividing streamline and associated field variables. These artifacts were more severe when isoparametric elements were used but disappeared when the local concentration of nodes in the mesh was raised.

Transient response. A more stringent test of mesh generation is the computation of transient free surface flows. Unstable steady states of large gap width (Fig. 7) were perturbed and their ensuing development followed in time. Figure 14 shows one before the pressure difference between upper meniscus and the lower meniscus was raised slightly, from 161.1 to 161.5 Pa (this perturbation is small and changes the spectrum only imperceptibly). In Fig. 15, the most dangerous part of the eigenvalue spectrum of the

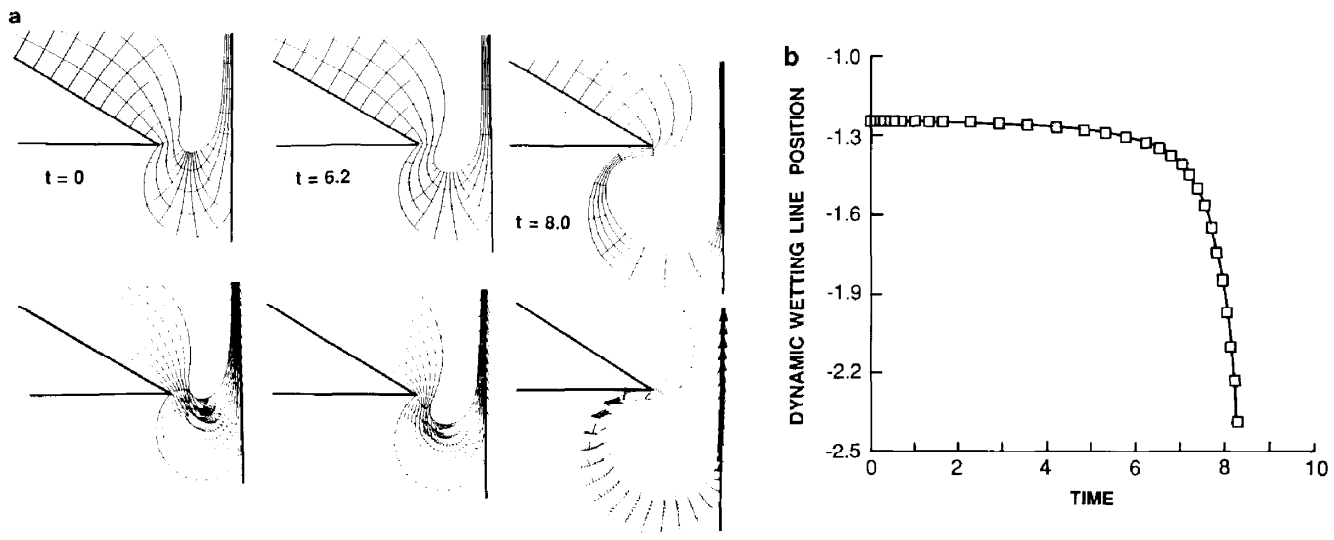

FIG. 15. (a) Transient response to a small increase of pressure difference between upper and lower free surface, from 161.1 to 161.5 Pa; (b) Dynamic wetting line position (dimensionless) as a function of time (dimensionless).

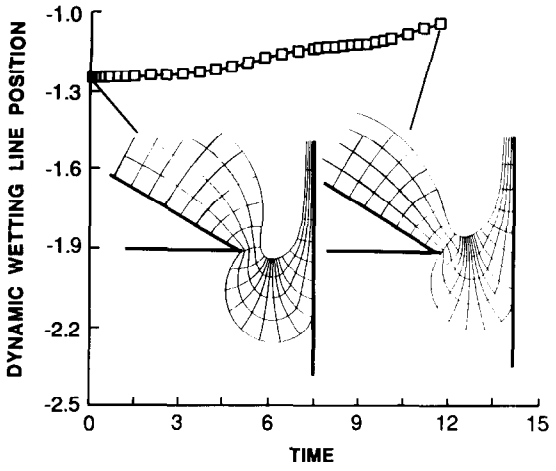

FIG. 16. Transient response to a small decrease of pressure difference across the liquid bridge, from 161.1 to 160.5 Pa.

steady state reveals that this steady state is unstable: there is a positive, real eigenvalue. The perturbed flow was followed until the lower meniscus became tangent to the lower lip, after eight time units (approximately 40 ms). Figure 15a shows the discretization and predicted velocity fields at selected times, and Fig. 15b shows the wetting line position as a function of time. The response to the slight pressure rise was exponential-like growth—ballooning—of the coating bead, which agrees with the results of linear stability analysis. The elliptically generated mesh promptly followed the rapidly rearranging flow, with no need of changing the control functions or the control parameters.

Figure 16 shows the wetting line position as a function of time after the pressure difference was reduced slightly from 161.1 to 160.5 Pa. The initial steady state is the same as in Fig. 14 and thus unstable. After 12 time units (60 ms), little change has taken place: the bead has thickened slightly! So the response depends critically on the direction of the initial disturbance, and elliptic mesh generation facilitates uncovering this asymmetry.

Disturbing the wetting line position—a direct perturbation of the state of the system—provides another fascinating example. The state was chosen to lie between the Hopf point and the turning point of the family of states of varying gap width (Fig. 7). After a short oscillatory transient the wetting line seems to come to rest (Fig. 17b), a response that appears discordant with linear analysis (in the spectrum of the steady state there are two positive, real eigenvalues; see Fig. 17a). Figure 18 shows that the apparent return to a steady state was actually part of a long incubation period of about 20 time units (100 ms). The chosen disturbance was poor in the unstable modes and richer in some stable oscillatory modes. Consequently it took a long time compared to the initial transient for the unstable modes to grow to discernible magnitude.

The preceding examples show the capabilities of the new elliptic system in discretizing entire families of steady states as well as rapidly rearranging transient states of free surface flows. They also permit critical testing of linear stability analysis [29] and they emphasize that it is more difficult to assess stability by transient simulation than by linear stability analysis.

5. DISCUSSION

The performance of the elliptic mesh generation system has been illustrated with examples of viscous free surface flows. The development was based on considerable experimentation that led not only to the design of partial differential equations and boundary conditions, as explained in Section 3, but also to the establishment of guidelines for the selection of control parameters and control functions, as described below.

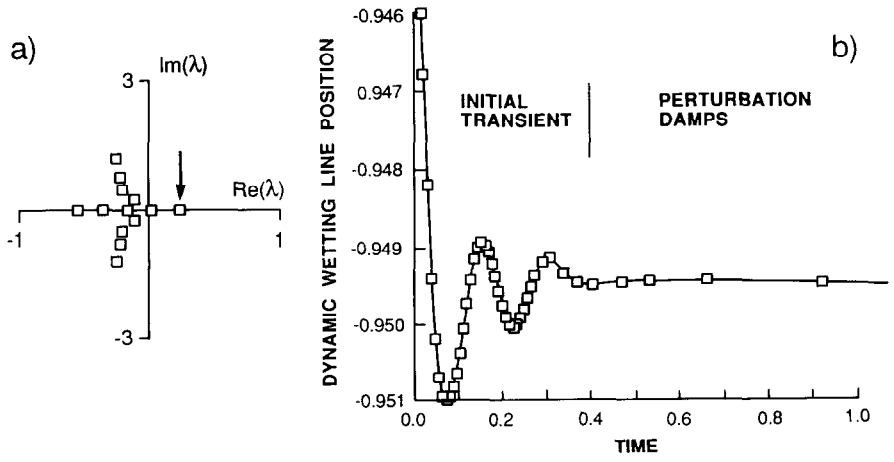

FIG. 17. Transient response to a small perturbation of dynamic wetting line position (short time scale): (a) most dangerous part of the spectrum; (b) dynamic wetting line position (dimensionless) as a function of time (dimensionless).

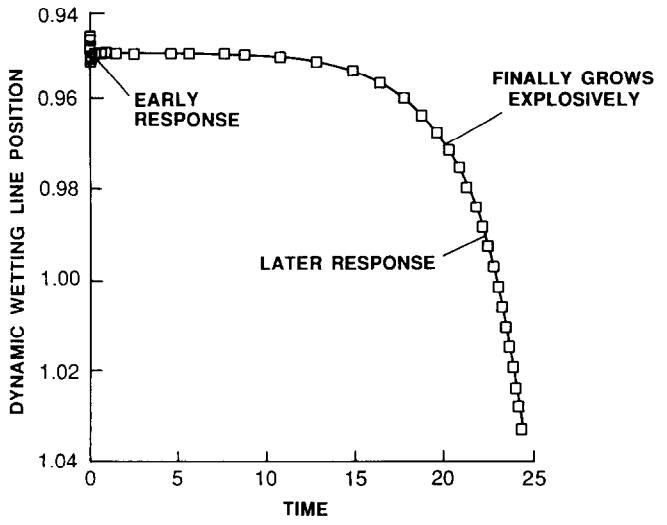

FIG. 18. Transient response to small perturbation of dynamic wetting line position (longer time scale): dynamic wetting line position (dimensionless) as a function of time (dimensionless).

5.1. *Inputs to the New Method*

By virtue of the dimensional homogeneity of individual terms in the mesh generating system (8) and (9), selection of control parameters is straightforward. ε_1 and ε_2 should normally be of order unity; values in the range 0.1–0.3 have worked well in most cases. M_1 and M_2 equal to 1000 have generally performed well. The precise values of these parameters are not crucial if the boundary has neither highly curved parts nor acute corner angles. Indeed, we have found that in such cases the smoothness parameter ε_s and the penalty parameters M_1 and M_2 can be set to zero: the resulting mesh is still nearly orthogonal and nodes are distributed as specified by the control functions $f(\xi)$ and $g(\eta)$. Only when highly concave boundaries or highly acute corner angles are present may nonzero values of ε_s (never larger than 0.1 in our experience) be necessary to avoid the crossing of coordinate curves that are transverse to those parts of the boundary.

Increasing ε_1 or ε_2, say by a factor 100, enforces the desired distribution along the corresponding coordinate direction. Setting M_1 or M_2 to a number of the order of 1000 accomplishes the same along the corresponding boundary segment. Doing both is useful when the mesh needs refinement near concave boundaries (e.g., Fig. 2), because requirements of orthogonality and smoothness dilute coordinate curves near such boundaries (cf. [5, p. 192]). Avoiding the dilution usually compromises orthogonality only a little, because even one of Eqs. (13), with the corresponding control parameter (ε_1 or ε_2) small, is enough to generate a nearly orthogonal mesh; this was demonstrated by Visbal and Knight [20] in their work on a related method.

Briefly, increasing ε_1, ε_2, or ε_s too much can result in severe departures from orthogonality at the corners of subregions. The lower limits on these weights are set by lack of convergence of the Newton iteration, owing to the infinity of solutions associated with the orthogonalization terms alone (cf. Eqs. (1)). Actually the abundance of solutions that maximize orthogonality appears to be an advantage; the probability of the mesh generation system not having a solution for a given set of control parameters and functions is small, and any reasonable estimate of a roughly orthogonal mesh is likely to fall within the range of convergence to a solution of (13) by the Newton iteration used.

The same control or distribution functions $f(\xi)$ and $g(\eta)$ were employed in the interior of the mesh and in boundary terms in (13) in order to avoid mesh boundary layers that could otherwise arise. Constant, linear, or hyperbolic sine functions were chosen here for simplicity. Similar distribution functions have been used commonly in algebraic mesh generation, as detailed for example by Vinokur [40]; other choices may be more appropriate in resolving features of other physical problems.

5.2. *Subregions and Patching*

The examples in Figs. 5, 9, and 10 are flow regions that all map to a single topologically quadrilateral domain. These regions are simply divided into sequential subregions of flow down the inclined slide, flow in the bridging bead, and flow on the departing substrate. The only differences are in the control functions $f(\xi)$ and $g(\eta)$ and the control parameters ε_1 and ε_2. Patching together these subregions is automatic when, as in these examples, the row count of elements does not change from one subregion to the next.

If the flow wets the underside of the die lip the physical domain must be mapped onto a patched computational domain like that shown in Fig. 12. The entire mesh is again generated simultaneously and large deviations from orthogonality are avoided when at least one of ε_1 and ε_2 is kept small. However, when one of these, say ε_2, is chosen large in order to achieve a desired distribution of nodes along level curves of ξ, and the two adjacent subregions differ significantly in size, distortion can arise at the common boundary; see Fig. 19a. This misbehavior is controlled by making ε_2 small at the nodes on the common boundary; see Fig. 19b. In this case the desired distribution along level curves of ξ is enforced independently in each subregion. Moreover, ε_2 can now be raised further to achieve near equidistribution; compare Fig. 19c. The same procedure was used in the case of the two-layer slide flow shown in Fig. 13.

5.3. *Conclusions and Future Work*

We have developed an elliptic mesh generation system for generating boundary-fitted, nearly orthogonal finite element discretizations of two-dimensional free and moving

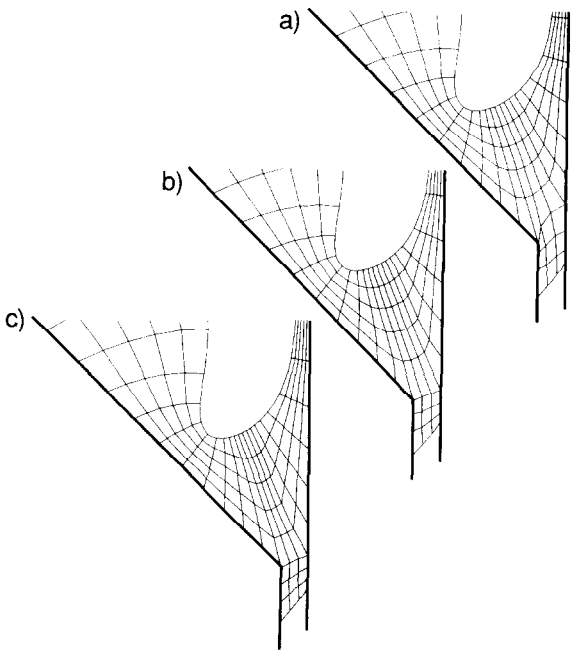

FIG. 19. Side-patching to generate a composite mesh: (a) $\varepsilon_1 = 0.25$, $\varepsilon_2 = 10\varepsilon_1$; (b) $\varepsilon_1 = 0.25, \varepsilon_2 = \varepsilon_1$; (c) $\varepsilon_1 = 0.25, \varepsilon_2 = 10\varepsilon_1, \varepsilon_2 = 0.01$ at nodes at boundary between subregions.

boundary problems. The superior capabilities of this system compared to algebraic mesh generation techniques have been demonstrated in computing entire families of steady states as well as rapidly evolving transient states of slide coating flows.

The subparametric mapping put forward in Section 4 points to possibilities of further hybridization of elliptic and algebraic mesh generation in order to lessen the computational cost. Elliptic mesh generation might be used to produce a coarse mesh of sufficient orthogonality, and a simple algorithm, say transfinite interpolation, could be used to generate on it a finer mesh (see, e.g., [14]).

The control or distribution functions employed in this work are not adaptive. To add adaptivity to the method the control functions can be made to depend on the solution of the physical problem [23, 42]. Preliminary results are encouraging in cases in which singularities are absent. However, in practically important free surface flows there are stress singularities at static and dynamic contact lines [43]. Further research is underway to identify effective indicators of solution error in such cases.

ACKNOWLEDGMENTS

We are grateful to H. K. Moffat for generously making available to us his version of the differential-algebraic system solver DASSL. We thank W. Van Abbenyen, A. Kovacs, and E. J. Lightfoot for their help. The work reported here was supported by grants-in-aid from Eastman Kodak Company and the Minnesota Mining and Manufacturing Company, and most recently through the Center for Interfacial Engineering. It was also supported by the University of Minnesota through a Graduate Dissertation Fellowship to K. N. Christodoulou and grants for computation from the Minnesota Supercomputer Institute.

REFERENCES

1. G. Strang and G. Fix, *An Analysis of the Finite Element Method* (Prentice–Hall, Englewood Cliffs, NJ, 1973).
2. W. C. Thacker, *Int. J. Numer. Methods Eng.* **15**, 1335 (1980).
3. J. K. Dukowicz, *J. Comput. Phys.* **54**, 411 (1984).
4. J. F. Thompson, Z. U. A. Warsi, and W. C. Mastin, *J. Comput. Phys.* **47**, 1 (1982).
5. J. F. Thompson, Z. U. A. Warsi, and W. C. Mastin, *Numerical Grid Generation* (Elsevier, New York, 1985).
6. R. E. Smith, *Appl. Math. Comput.* **10-11**, 137 (1982).
7. S. F. Kistler and L. E. Scriven, in *Computational Analysis of Polymer Processing*, edited by J. R. A. Pearson and S. M. Richardson (Appl. Sci. Essex, UK, 1983), p. 244.
8. K. N. Christodoulou and L. E. Scriven, *J. Fluid Mech.* **208**, 321 (1989).
9. D. Kerlick and G. H. Klopfer, *Appl. Math. Comput.* **10-11**, 787 (1982).
10. P. R. Eiseman, *Appl. Math. Comput.* **10-11**, 193 (1982).
11. D. E. Potter and G. H. Tuttle, *J. Comput. Phys.* **13**, 483 (1973).
12. S. B. Pope, *J. Comput. Phys.* **26**, 197 (1978).
13. S. K. Godounov and G. P. Prokopov, *USSR Comput. Math. Math. Phys.* **7**, 89 (1967).
14. P. Morice, in *Advances in Grid Generation*, FED-Vol. 5, edited by K. N. Ghia and U. Ghia (ASME Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, 1983), p. 71.
15. G. Ryskin and L. Leal, *J. Comput. Phys.* **50**, 71 (1983).
16. S. K. Godounov, *Actes Congr. Int. Math.* **3**, 99 (1970).
17. P. D. Thomas and J. F. Middlecoff, *AIAA J.* **18**, 632 (1980).
18. P. D. Thomas, *AIAA J.* **20**, 1195 (1982).
19. S. P. Shanks and J. F. Thompson, in *Proceedings, 2nd Int. Conf. on Numerical Ship Hydrodynamics, Berkeley, CA, 1977* (Univ. of Calif. Extension Publs., Berkeley, 1977), p. 202.
20. M. Visbal, *Generation of Nearly-Orthogonal Body-fitted Coordinate Systems in Two-dimensional Singly-connected Regions*, M. S. thesis (Rutgers University, New Brunswick, NJ, 1980).
21. M. Visbal and D. Knight, *AIAA J.* **20**, 305 (1982).
22. J. Brackbill and J. Saltzman, *Appl. Math. Comput.* **10-11**, 865 (1982).
23. J. U. Brackbill, *Appl. Math. Comput.* **10-11**, 277 (1982).
24. J. S. Saltzman and J. U. Brackbill, *J. Comput. Phys.* **46**, 342 (1982).
25. R. I. Kreis, F. C. Thames, and H. A. Hassan, *AIAA J.* **24**, 404 (1986).
26. R. Carcaillet, G. S. Dulikravich, and S. R. Kennon, *Comput. Methods Appl. Mech. Eng.* **57**, 279 (1986).
27. K.-J. Bathe, *Finite Element Procedures in Engineering Analysis* (Prentice–Hall, Englewood Cliffs, NJ, 1982).
28. K. N. Christodoulou and L. E. Scriven, Stability of two-dimensional viscous free surface flows to small three-dimensional disturbances, in preparation.
29. K. N. Christodoulou and L. E. Scriven, *J. Sci. Comput.* **3**, 355 (1988).
30. P. Loetstedt, *SIAM J. Appl. Math.* **42**, 281 (1982).
31. P. Loetstedt, *SIAM J. Sci. Stat. Comput.* **5**, 370 (1984).
32. P. S. Huyakorn, C. Taylor, R. L. Lee, and P. M. Gresho, *Comput. Fluids* **6**, 25 (1978).
33. K. J. Ruschak, *Int. J. Numer. Methods Eng.* **15**, 639 (1980).

34. L. R. Petzold, *SIAM J. Sci. Stat. Comput.* **3**, 367 (1982).
35. P. Hood, *Int. J. Numer. Methods Eng.* **10**, 379 (1976).
36. L. R. Petzold, SAND 82-8637, Sandia National Laboratories, Livermore, CA, 1982 (unpublished).
37. L. R. Petzold and P. Loetstedt, *SIAM J. Sci. Stat. Comput.* **7**, 720 (1986).
38. I. Christie, D. F. Griffiths, A. R. Mitchell, and O. C. Zienkiewicz, *Int. J. Numer. Methods Eng.* **10**, 1389 (1976).
39. W. Van Abbenyen, K. N. Christodoulou, and L. E. Scriven, Frequency response of coating flows, submitted.
40. M. Vinokur, *J. Comput. Phys.* **50**, 215 (1983).
41. R. Loehner, K. Morgan, J. Peraire, and O. C. Zienkiewicz, in *The Free-Langrange Method*, edited by M. J. Fritts, W. P. Crowley, and H. Trease (Springer-Verlag, New York/Berlin, 1985), p. 236.
42. L. Demkowicz and J. T. Oden, *Int. J. Eng. Sci.* **24**, 55 (1986).
43. C. Huh and L. E. Scriven, *J. Colloid Interface Sci.* **35**, 85 (1971).