JOURNAL OF COMPUTATIONAL PHYSICS **125,** 262–276 (1996)
ARTICLE NO. 0092

Numerical Solution of Free-Boundary Problems: Calculation of Interface Evolution during CVD Growth

JACOB J. THIART AND VLADIMIR HLAVACEK*

Laboratory for Ceramic and Reaction Engineering, Department of Chemical Engineering, C. C. Furnas Hall, State University of New York at Buffalo, Amherst, New York 14260

Received November 14, 1994; revised July 3, 1995

A procedure is described for numerical solution of free-boundary problems where highly nonlinear boundary shapes develop. A simple explicit solution method is used, and special attention is given to two aspects. The first is the introduction of a parametrization, which *stretches* the boundary/interface and allows tracing the development of interface shapes beyond the point where a single-valued representation becomes inappropriate. The second is a procedure for automatic generation of a mesh for discretizing the domain above a highly nonlinear interface. Examples of the evolution of a gas–solid interface during solid film growth by chemical vapor deposition are used to illustrate application of the solution method. The effect of different values of mesh generation parameters on placement of nodes are shown, as well as the effect of the number of nodes along the interface. It was found that in cases with a highly distorted boundary, the shape of the boundary can be used in the weight functions to ensure appropriate concentration of nodes in groove regions. The number of nodes along the interface is less crucial than the choice of interface *length* to be studied. © 1996 Academic Press, Inc.

1. INTRODUCTION

Free-boundary problems exist in a wide variety of systems and have received much attention, due to their inherent mathematical complexity. Numerical solution of these systems are complicated by the unknown shape and position of one domain boundary, typically an interface between two distinct phases. Examples of such systems include free surface waves [1], two-phase flow in porous media [2], flow in a Hele–Shaw cell [3], solidification from a melt [4], flame propagation in gas mixtures [5] and solid–solid combustion [6], growth of a solid film during physical vapor deposition (PVD) [7], or chemical vapor deposition (CVD) [8].

The present paper describes a procedure for numerical solution of free-boundary problems, as applied to solid film growth during CVD. We restrict our attention to Stefan-like problems [9], in which there is one field variable and field equations are linear. In the case of isothermal CVD growth, this field variable is gas phase reactant concentration. The Galerkin finite element method is used to solve the domain problem, while finite difference techniques are used for time stepping and to account for the changing shape of the moving boundary. Highly nonlinear or distorted gas–solid interface shapes are possible, which necessitates the use of an appropriate interface representation and introduces complications with discretization of the two-dimensional (2D) domain above it. We therefore focus on two aspects of the numerical solution procedure, namely the choice of an appropriate parametrization for representing the interface and an automatic mesh-generation scheme for discretizing the 2D domain above the interface. Several examples are used to illustrate application of the numerical solution procedure.

2. BACKGROUND

Several approaches have been used in the numerical solution of Stefan-like problems. One of the earliest was relaxation methods used for determining percolation of fluid through porous media, as described by Southwell [10]. The position of the moving boundary was determined iteratively by using one boundary condition as a *predictor* and the other in the *corrector* step.

The boundary integral technique, which uses Green's theorem to relate changes in the domain to changes along the moving boundary [1, 11], has the advantage that the position of the boundary can be determined without having to discretize the entire domain, and the dimensionality of the problem is reduced by one, resulting in fewer unknowns. However, integral methods are generally computationally more demanding than difference methods, since matrices to be inverted are not sparse or banded. Other approaches are based on the classical finite element (or finite difference) methods for solving field equations on a discretized domain with appropriate boundary conditions. The difference is that in free-boundary problems this domain changes shape, since one of its boundaries is the moving interface. In addition to the usual boundary condition at the moving interface, another relation is required

* Correspondence author.

0021-9991/96 $18.00

to account for its changing shape. The solution is obtained in one of the following ways:

The first option is to map the domain with changing shape onto a fixed domain by transforming the field equations. The problem is then solved by, e.g., the Galerkin finite element method in the transformed domain, and the solution is transformed back to the original domain. This method has the advantage that mesh generation itself is simple and needs to be performed only at the start of a solution run. The disadvantage is that the transformed field equations can become significantly more complicated.

The second approach is to use a mesh which changes shape with the changing interface and to solve the original problem in the adaptive mesh. No transformation of field equations is necessary, but the mesh shape has to be continually updated to follow the shape of the moving boundary, which can cause computational difficulties.

Christodoulou and Scriven [12] reviewed different algebraic and elliptic mesh generation methods, including conformal mapping, orthogonal mapping, and variational methods, such as that of Brackbill and Saltzmann. They highlighted some drawbacks to these methods and proposed a new elliptic mesh generation method, which combines optimization of the mesh with an orthogonality measure directly related to the discretization error. Tsiveriotis and Brown [13] pointed out that the resulting equations were not fundamentally different than those derived by Brackbill and Saltzmann and that the adaption part of the equations were hyperbolic, leading to an ill-posed boundary value problem for high parameter values. Tsiveriotis and Brown then proposed a two-step mapping procedure called the "mixed mapping method" (MMM), which is particularly suited to mapping the region inside highly deformed cells. The method was shown to overcome the problems associated with other variational methods.

Ettouney and Brown [14] reviewed different numerical solution approaches for Stefan-like solidification problems, both linear and nonlinear. They found that the Galerkin finite element method is ideally suited to solving Stefan-like problems, since the flux boundary conditions at the moving interface are easily incorporated as natural conditions for the Galerkin residual equations. They also found that the choice of boundary condition *distinguished* for locating the position of the interface played an important role in determining the accuracy of results.

The present study deals with unsteady growth of solid films from the vapor phase, an initial-boundary value problem. We illustrate the use of an explicit solution scheme, as applied to the evolution of highly nonlinear interface shapes, such as those observed during diffusion-limited CVD growth. The field equation (with reactant concentration as field variable) is solved in the physical coordinates by the Galerkin finite element method. The field equation therefore is not transformed, but an adaptive mesh is updated at each time step to trace the changing shape of the interface. In addition, since we consider cases where interface shape is not a single-valued function of one spatial coordinate, a parametrization is introduced which *stretches* the interface and allows tracing highly nonlinear shapes without difficulty. The choice of an appropriate parametrization is an important part of the solution procedure and will be discussed in more detail later.

3. MODEL OF INTERFACE EVOLUTION DURING CHEMICAL VAPOR DEPOSITION (CVD)

Consider the growth of an amorphous solid film during a typical high pressure (1 atm) CVD process. The modeling setup is shown in Fig. 1. Reactants diffuse through a stagnant boundary layer to the gas–solid interface, where adsorption and desorption take place. Reactants are converted to solid product by a chemical reaction, which is assumed to be irreversible and first order with respect to the limiting reactant. In addition, surface diffusion of deposited solid material takes place. This phenomenon tends to minimize surface energy and therefore flattens protrusions and fills up grooves. It can be regarded as a stabilizing effect for planar film growth. On the other hand, the shorter diffusion length to tips of protrusions tends to enhance growth there and destabilizes planar film growth. A balance between these opposing effects leads to the development of a characteristic interface shape.

The modeling and computational domain is the stagnant gas boundary layer above the moving interface. A more detailed description of the three-dimensional (3D) model formulation is given in Viljoen *et al.* [8]. The two-dimensional (2D) model can be summarized as follows (in dimensionless form):

The **gas phase** governing equation:

$$\frac{\partial \mathscr{C}}{\partial t} = \frac{1}{\mathrm{Pe}}(\nabla^2 \mathscr{C}) \tag{1}$$

with boundary conditions

$$\begin{aligned}
\mathscr{C} = 1 \quad &\text{at } z = \frac{1}{\eta} + t,\\
(\nabla \mathscr{C} \cdot \mathbf{n}) = \eta\,\mathrm{Da}\,\mathscr{C}(1+\kappa) \quad &\text{at } z = \mathscr{H},\\
\frac{\partial \mathscr{C}}{\partial x}\Big|_{x=0} = \frac{\partial \mathscr{C}}{\partial x}\Big|_{x=nl/\Gamma},&\\
\mathscr{C}(0, z) = \mathscr{C}\left(\frac{nl}{\Gamma}, z\right) \quad &\text{with } n = 0, \pm 1, \pm 2, \ldots,
\end{aligned} \tag{2}$$

where κ represents the dimensionless curvature and the dimensionless parameters are defined as

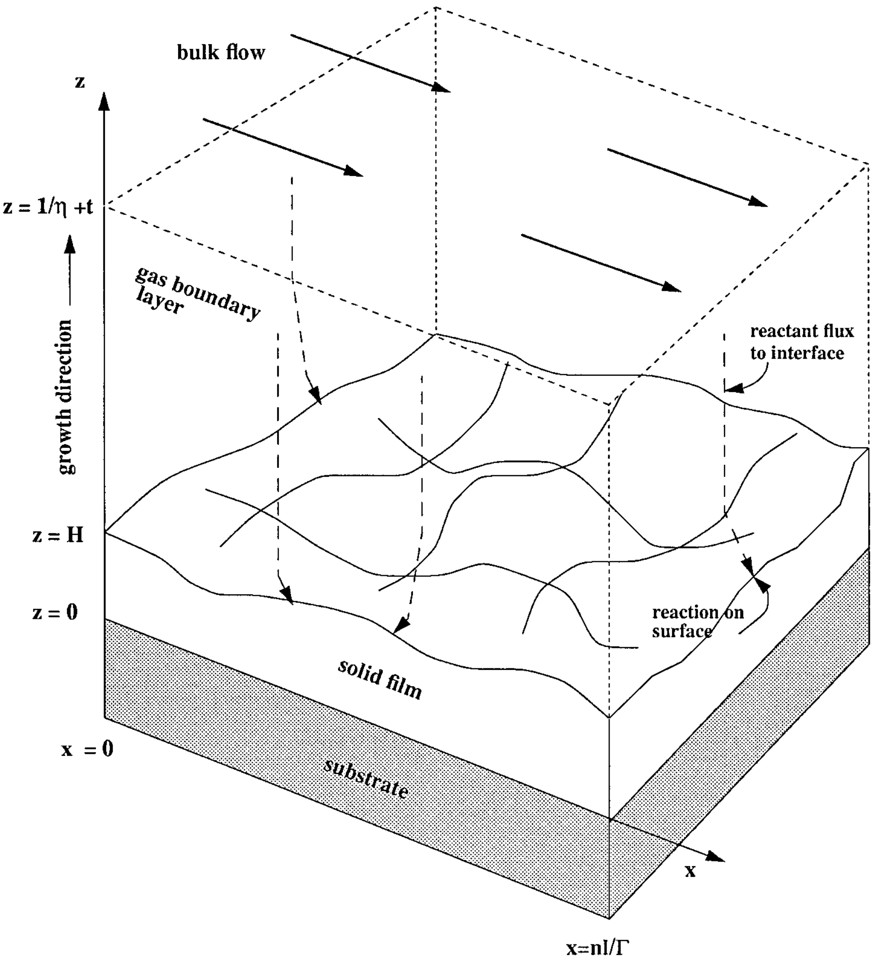

FIG. 1. Modeling setup of growing film during CVD.

$$\mathrm{Pe} = \frac{V_0\Gamma}{D_f}, \quad \mathrm{Da} = \frac{K\delta}{D_f}, \quad \eta = \frac{\Gamma}{\delta}.$$

Other symbols are defined in the Appendix.

An additional boundary condition is necessary to calculate the shape and position of the interface and in the present case it is a mass balance in the solid phase at the moving interface. The **solid phase** equation expressed in terms of local interface velocity, v, takes the form

$$v = \frac{\beta C_b \eta\, \mathrm{Da}}{\mathrm{Pe}} \mathscr{C}(1+\kappa) - \frac{\phi}{\mathrm{Pe}} \frac{\partial^2 \kappa}{\partial s^2}, \qquad (3)$$

where

$$\phi = \frac{D_s^*}{D_f \Gamma}, \quad D_s^* = D_s \Omega \alpha.$$

The parameter κ represents interface curvature, and s is the arc-length along the interface. An important characteristic of Eq. (3) is that the term $\partial^2\kappa/\partial s^2$ represents a highly nonlinear fourth-order spatial derivative, as Eq. (5) in the next section will show. This is not common for diffusion-like terms, which are usually second-order and can cause problems with discretization, as well as computational difficulties.

4. NUMERICAL SOLUTION

A time scale analysis of the system reveals that the characteristic time for gas diffusion is much shorter than that for interface evolution, which suggests that the gas phase is in pseudo-steady state with respect to changes in interface shape [15]. Therefore, the problem is followed on the time scale of interface evolution and the concentration field is determined as steady state at each time step. Solution of the field equation and interface shape is performed in *successive* steps, therefore, by an explicit method.

4.1. *Choice of Parametrization*

In cases where the surface remains fairly smooth and no deep grooves develop, a single-valued representation of the interface may be used. The interface is expressed as a function of a parameter, σ, where the parameter is chosen as the horizontal coordinate, x. A geometric conversion is used to convert normal growth to vertical displacement. In the case of CVD growth, let the flux of solid material contributing to film growth be $\gamma(\mathcal{H}, \mathcal{C})$, and the effect of surface diffusion $\chi(\mathcal{H})$, where these terms can depend on interface shape, $\mathcal{H}$, and concentration at the interface, $\mathcal{C}$. Then

$$z_t = \sqrt{1 + z_x^2}\,(\gamma(\mathcal{H}, \mathcal{C}) + \chi(\mathcal{H})), \tag{4}$$

where subscripts denote derivatives with respect to that variable. For free-boundary systems, in general, $\gamma(\mathcal{H}, \mathcal{C})$ can be thought of as any phenomenon contributing to translation of the interface normal to its previous position, which can depend on distribution of the field variable at the interface and the shape of the interface. In solidification problems, for example, it would be a thermal gradient or flux of heat. The term $\chi(\mathcal{H})$ represents a surface tension term, which usually depends on local curvature of the interface and tends to limit the size of protrusions and minimize surface energy. In the present solid growth system, the effect of surface diffusion is proportional to the second derivative of curvature with respect to arc-length along the curve [16]; thus $\chi \propto \partial^2\kappa/\partial s^2$. In terms of x and z,

$$\frac{\partial^2\kappa}{\partial s^2} = -\frac{1}{\sqrt{1 + z_x^2}}\left(\frac{\kappa_x}{\sqrt{1 + z_x^2}}\right)_x, \tag{5}$$

where interface curvature is defined as

$$\kappa = \frac{z_{xx}}{(1 + z_x^2)^{3/2}}.$$

It is clear that this formulation becomes inappropriate when interface gradients become very large; thus when $z_x \to \infty$. In the present study we will illustrate the use of a more appropriate choice of parameter, which will permit the tracing of highly nonlinear interface shapes. For initial or boundary value problems where steep gradients and large curvatures are encountered, e.g., in flame and combustion problems, the so-called *arc-length strategy* has typically been used [17, 18]. In this technique, a function, such as the position of the interface, is not expressed in terms of the horizontal coordinate, since this can lead to problems in regions of steep gradients. Instead, the function is expressed in terms of its change along the arc-length of the curve, which stretches the curve and eliminates steep gradients. A multi-valued function in the horizontal coordinate becomes a single-valued function in the arc-length of the curve. Therefore, the arc-length (say s) is introduced as a new independent variable and the original independent variable (say x) becomes a dependent variable. The relation between x and s introduces an additional equation and the original system of equations are transformed and expressed in terms of the arc-length s. The problem is solved in terms of the new parameter s, which proves computationally more convenient in the region of steep gradients than the original system. The method has proved to be a very useful tool in continuation methods and, e.g., for tracing solution branches in bifurcation diagrams [19].

The arc-length seems an appropriate parameter for expressing interface position during CVD growth. However, in an initial-boundary value problem it has the disadvantage that the total arc-length changes with time and its value at each point in time is not known a priori. An appropriate choice of parameter is one which has a fixed total length independent of time and which can be discretized into an equidistant grid. The *normalized* arc-length satisfies all the requirements, since its range is constant and known a priori (it is chosen arbitrarily), and the position of the interface is a unique function of σ. This means that the gradients $\partial x/\partial\sigma$ and $\partial z/\partial\sigma$ cannot approach infinity as in the former case where σ was chosen as x. As a simple example, consider the interface shown in Fig. 2a in the conventional Cartesian representation. Clearly the interface cannot be expressed as a function of either coordinate. By introducing the normalized arc-length as new independent variable, the interface is *stretched* and its horizontal and vertical coordinates become single-valued functions of the new parameter, σ (see Figs. 2b and 2c).

In the numerical solution procedure, the interface is discretized into an equidistant grid with a predetermined number of points and spatial derivatives at each point are represented by appropriate finite difference formuli. The relation between σ and the position of the interface is not an analytical function, since the interface position is known only at discrete values of the parameter, but this is not a limitation. All the important quantities can be expressed in discrete form in terms of the parameter, σ. The first is the arc-length s, which is needed in the determination of the effect of surface diffusion. It can be defined as

$$s(\sigma) = \int_0^\sigma \sqrt{x_\varsigma^2 + z_\varsigma^2}\, d\varsigma.$$

Since growth is normal to the interface, it is also important to have expressions for the tangential and normal unit vectors. These are

$$\mathbf{t} = \left(\frac{x_\sigma}{\sqrt{x_\sigma^2 + z_\sigma^2}}; \frac{z_\sigma}{\sqrt{x_\sigma^2 + z_\sigma^2}}\right)$$

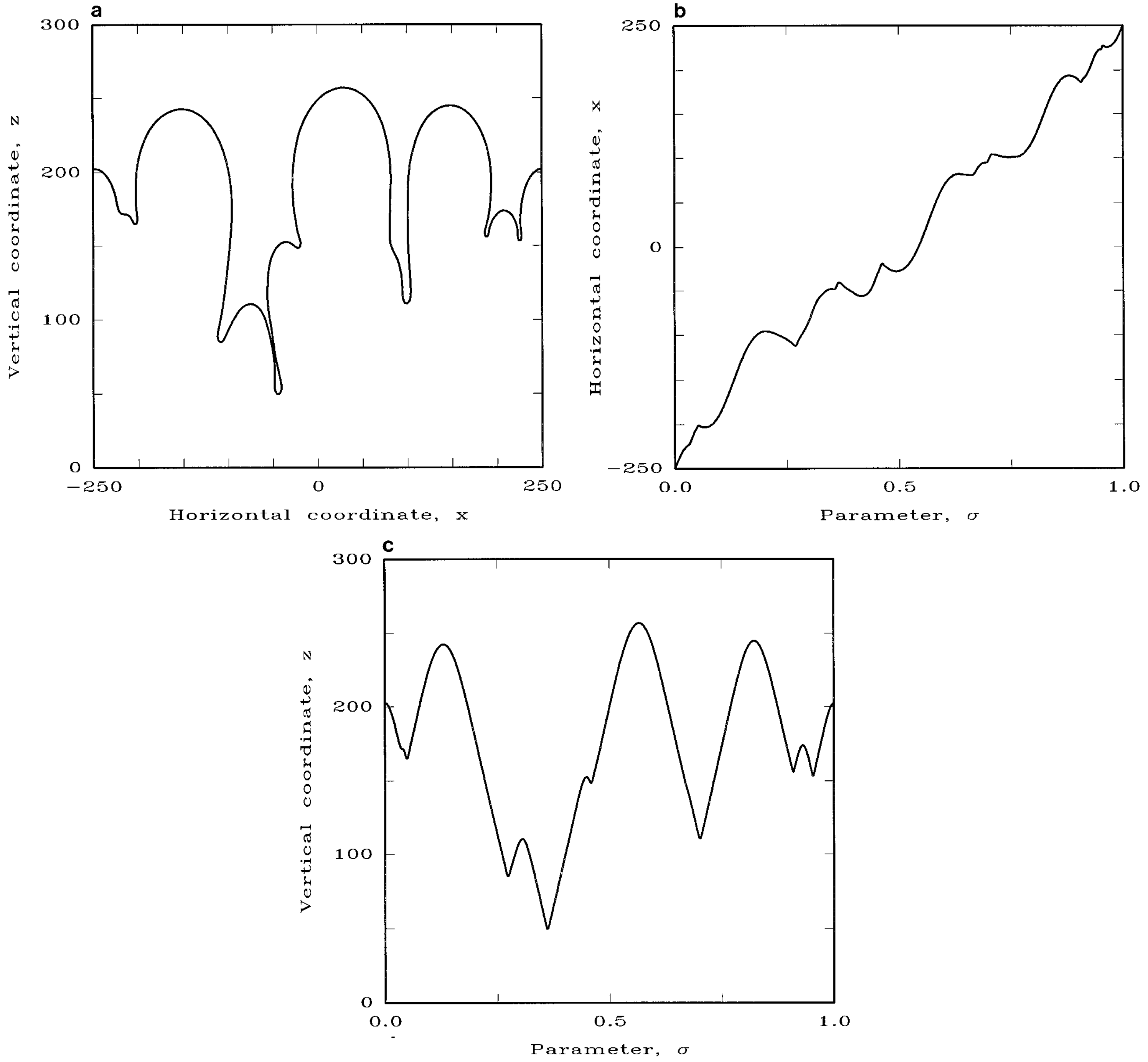

FIG. 2. Example of stretching interface and transforming a multi-valued function into a single-valued representation by introducing normalized arc-length as parameter: (a) Cartesian representation; (b), (c) parametric form.

$$\mathbf{n} = \left(\frac{-z_\sigma}{\sqrt{x_\sigma^2 + z_\sigma^2}}; \frac{x_\sigma}{\sqrt{x_\sigma^2 + z_\sigma^2}} \right).$$

The curvature is defined as

$$\kappa = |d\mathbf{t}/ds|,$$

which is equivalent to

$$\kappa = \frac{x_\sigma z_{\sigma\sigma} - z_\sigma x_{\sigma\sigma}}{(x_\sigma^2 + z_\sigma^2)^{3/2}}.$$

The change in position of the interface is now given by

$$\frac{\partial x}{\partial t} = \frac{-z_\sigma}{\sqrt{x_\sigma^2 + z_\sigma^2}} (\gamma(\mathscr{H}, \mathscr{C}) + \chi(\mathscr{H})) \tag{6}$$

$$\frac{\partial z}{\partial t} = \frac{x_\sigma}{\sqrt{x_\sigma^2 + z_\sigma^2}} (\gamma(\mathscr{H}, \mathscr{C}) + \chi(\mathscr{H})), \tag{7}$$

where $\gamma(\mathscr{H}, \mathscr{C})$ and $\chi(\mathscr{H})$ are now expressed in terms of the parameter σ. All derivatives are also with respect to σ, so an appropriate choice of σ is one where x_σ and z_σ remain finite at all times.

4.2. *Automatic Mesh Generation*

The present study deals with the solution of field equations in a domain which continuously changes shape, due to the changing shape of one of its boundaries. Instead of mapping this deforming domain onto a fixed domain, the problem is solved in the original domain. This necessitates the generation of a new mesh with each step in time, which can be very time-consuming. It is therefore necessary to use an efficient automatic mesh generation scheme.

The mesh generation problem requires the placement of nodes inside a domain and along its four boundaries in such a way that the resulting elements have shapes and connectivity which will ensure an appropriate finite element representation. The aim is to obtain quadrilateral cells as close as possible to rectangles (no angles close to 0° or 180°) with aspect ratios preferably smaller than 10 to 1. Three boundaries of the domain are straight lines and the fourth coincides with the moving interface, which can assume virtually any shape. At each point in time the location of all four boundaries will be known, and mesh generation becomes a problem of distributing interior nodes in an appropriate way. Accordingly, one can define three optimization criteria for placement of interior nodes. These are: (i) adaption, (ii) smoothness, and (iii) orthogonality. The first criterion refers to concentrating nodes in specific regions inside the domain, e.g., close to one of the boundaries. This is done to ensure sufficient resolution in areas where severe changes or steep gradients are anticipated, such as is the case in combustion problems [20]. For CVD growth, reactant concentration close to the interface is important for determining interface dynamics and it is desirable to put more nodes close to this boundary. In addition, the number of points along the interface may be larger than that in the vertical direction (up to 10 to 1), which means strong adaption is necessary at the interface to ensure acceptable aspect ratios of elements in that region.

Smoothness refers to the requirement that there should be a gradual change in cell size, since any rapid variations lead to inaccuracies [21]. The orthogonality criterion strives to make all cell angles 90°, which improves solution accuracy. In determining the position of interior nodes, the mesh generation scheme optimizes measurable quantities of these three criteria, expressed in terms of the positions of nodes. Different scalar weights are assigned to each criterion, depending on its importance, and these are combined into one objective function.

The procedure is essentially a discrete version of the variational method proposed by Brackbill and Saltzmann [22]. In the latter, functionals are introduced that provide measures of grid smoothness,

$$I_{\mathrm{s}} = \int_D [(\nabla\xi)^2 + (\nabla\zeta)^2]\, dV,$$

grid orthogonality,

$$I_{\mathrm{o}} = \int_D [(\nabla\xi\cdot\nabla\zeta)^2 J^3\, dV,$$

and weighted volume variation,

$$I_{\mathrm{v}} = \int_D wJ\, dV.$$

The mesh is a mapping of the physical x and z coordinates onto a computational domain and ξ and ζ, where $J = x_\xi z_\zeta - x_\zeta z_\xi$ represents the Jacobian of the mapping. The functional I_{v} measures the global property of adaption as represented by the weight function $w = w(x, z)$. Minimization of I_{v} results in small cell sizes wherever w is large, which ensures appropriate adaption. Integrals I_{s} and I_{o} ensure there are no rapid variations in cell sizes and keep grid lines close to orthogonal. A weighted average of the three functionals are used,

$$I = I_{\mathrm{s}} + \lambda_{\mathrm{v}} I_{\mathrm{v}} + \lambda_{\mathrm{o}} I_{\mathrm{o}},$$

where the scalar weights λ_{v} and λ_{o} provide a means to control the relative importance of each of the mesh properties. Application of the Euler–Lagrange equations for the minimization of I yields a nonlinear system of coupled elliptic partial differential equations, which have to be solved for the x and z positions of the grid points.

In the discrete method, formulation of the adaptive mesh generator can be considered as minimization of an objective function E [20, 23], which is the discrete form of I and depends on all grid positions $x_{i,j}$ and $z_{i,j}$. The condition for the minimum is

$$0 = dE = \sum_{j=1}^{M+1} \sum_{i=1}^{N+1} \left[\frac{\partial E}{\partial x_{i,j}} dx_{i,j} + \frac{\partial E}{\partial z_{i,j}} dz_{i,j} \right], \tag{8}$$

resulting in the mesh generator equations for each point,

$$\frac{\partial E}{\partial x_{i,j}} = 0, \quad \frac{\partial E}{\partial z_{i,j}} = 0.$$

These equations are nonlinear in the node coordinates x and z and have to be solved iteratively. Implementation of this method is computationally very expensive as a result of inversion of large matrices. Fortunately, determination

TABLE I

Computational Requirements for Different Parts of the Solution Procedure

Solution component	Time units (normalized) per iteration
Mesh generation	2.4
Solution of field equation	36
Time integration, recordkeeping	1.0

of the exact location of the grid points is not crucial, as long as critical parts of the physical solution are well resolved. This means one can use fewer iterations than is necessary for convergence. The mesh also does not change much from one time step to the next, which allows the use of simpler approximations, such as solution by point-Jacobi iterations or the ADI (alternating direction implicit) method [23]. Both simplifications are computationally much less demanding, but they lead to reduced coupling between grid points, which means more iterations are necessary. It takes several iterations before a point is affected by what happens on the other side of the grid. In our system, generation of the initial mesh at $t = 0$ may require 100 ADI iterations. However, since the mesh changes only slightly at every time step, a typical number of iterations required at $t > 0$ is 20. Table I gives an indication of the computational time requirements of different parts of the solution procedure. Assume for one complete cycle that we need the following number of iterations: 20 for mesh generation, 1 for solution of the field equation, and 20 for time integration. This means that the computational time will be divided as follows: 46% for mesh generation, 35% for solution of field equation, and 19% for time integration.

The following examples illustrate application of the automatic mesh generator and show the effect of adaption and orthogonality. Figure 3 shows three meshes generated for the domain above a simple interface shape. The ADI method is used to determine the position of the interior nodes on a 50 × 20 element mesh. Adaption is accounted for by the weight function, $w(x, z)$, and orthogonality by the scalar weight, λ_o, while smoothness is not explicitly accounted for by a separate scalar weight. It does, however, enter into the determination of $w(x, z)$ by using an exponential form where the magnitude of the exponent is controlled by the user. The parameters chosen for generating the mesh in Fig. 3a are

$$w(x, z) = 1 + \lambda_v \exp(-\tau\mu)$$
$$\tau = 0$$
$$\lambda_v = 300$$
$$\lambda_o = 3,$$

where the parameter μ varies linearly from $\mu = 0.0$ to $\mu = 1.0$ along the lines connecting nodes on the interface with those on the top boundary. This mesh can be considered a base case, since nodes are not concentrated in any specific area. The distribution of points seems very good and the distortion of elements not very significant. The orthogonality of lines can be increased by increasing the value of λ_o and decreasing λ_v, which results in the mesh shown in Fig. 3b. The parameters are

$$w(x, z) = 1 + \lambda_v \exp(-\tau\mu)$$
$$\tau = 0$$
$$\lambda_v = 3$$
$$\lambda_o = 15.$$

The orthogonality of lines has indeed improved, but the distribution of points is less ideal. In addition, fewer nodes are placed close to the interface than in the previous mesh. In CVD growth cases, it is important to place more nodes close to the curved boundary (which represents the gas–solid interface), since the concentration field in this region is crucial to determining the dynamics of the interface. A good balance between adaption and orthogonality is obtained by the set of values

$$w(x, z) = 1 + \lambda_v \exp(-\tau\mu)$$
$$\tau = 2$$
$$\lambda_v = 100$$
$$\lambda_o = 15,$$

as shown by the mesh in Fig. 3c. More nodes are placed close to the interface, and the distribution of points seems good. A simple quantitative measure of how closely nodes follow the curved boundary is the aspect ratio of elements close to or at that boundary. The larger the aspect ratio, the closer the nodes are to the boundary. For the three meshes in Fig. 3, Table II shows the difference in average aspect ratio at the curved boundary.

Now consider an example where the interface shape is more distorted. With parameters identical to those in Fig.

TABLE II

Comparison of Aspect Ratios for Elements at Curved Boundary

Mesh	Average aspect ratio
Fig. 3a	1.94
Fig. 3b	3.43
Fig. 3c	6.12

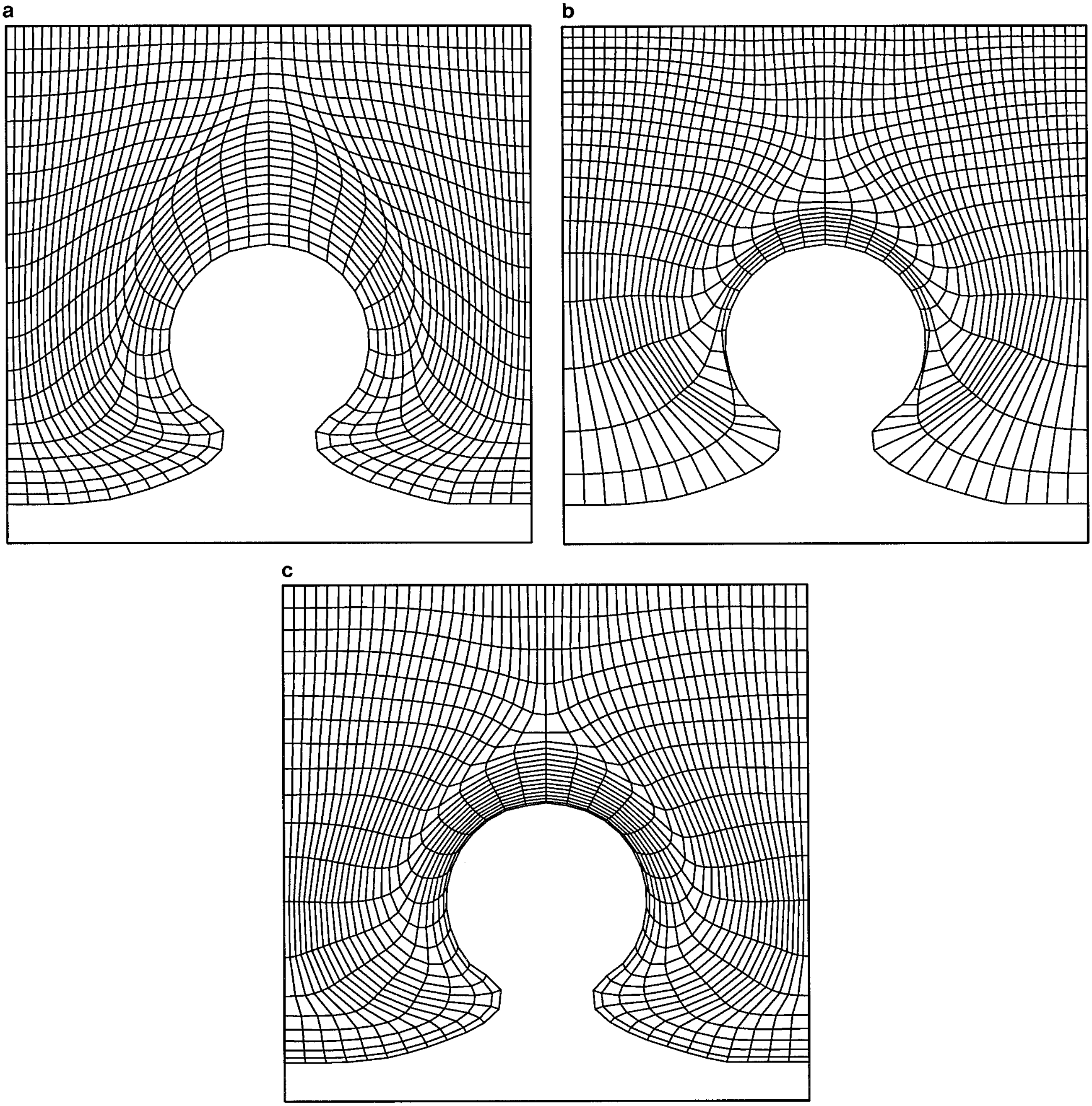

FIG. 3. Effect of orthogonality and different weights of adaption on mesh generation.

3c and a 100 × 20 element mesh, the result is shown in Fig. 4a. It is clear that a set of parameters which resulted in a well-distributed mesh for a simple interface shape, does not necessarily work well for a more demanding case. Although the overall distribution of points is good, the nodes do not trace the interface shape closely enough in deep grooves. It is therefore desirable to be able to identify regions with deep grooves and to adjust the weight functions accordingly. The examples so far have not taken account of the actual shape of the interface, only of the relative position of a node along the lines connecting the interface to the top of the domain. Let us attempt to concentrate nodes in areas of deep grooves by using the normalized height of the interface in the adaption function. Therefore, use a function h, where $h = 1.0$ at the bottom of the deepest groove and $h = 0.0$ at the top of the highest protrusion. As an extreme case, consider the following choice of parameters:

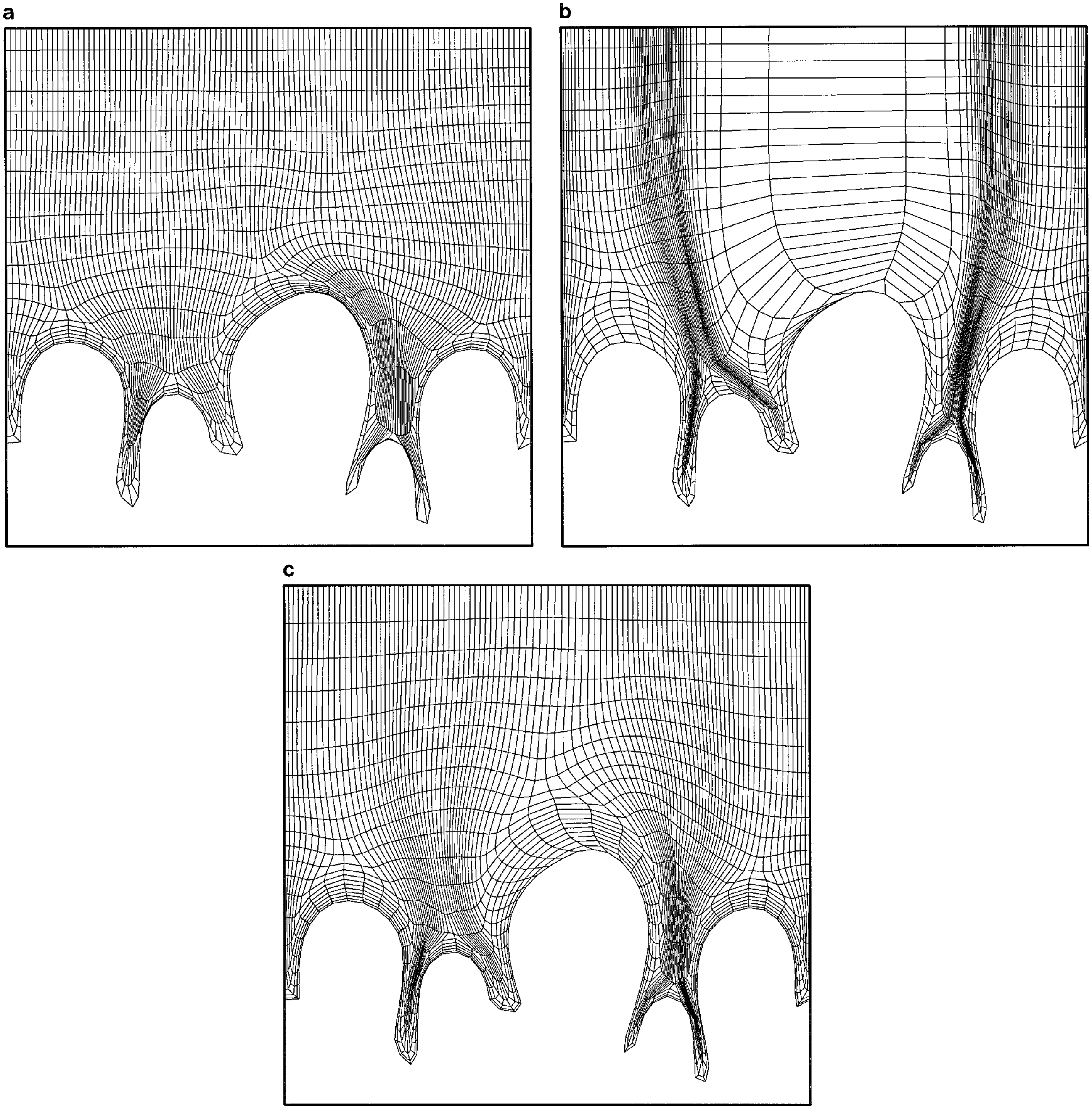

FIG. 4. Examples of mesh generation when interface shape is highly nonlinear.

$$w(x, z) = 1 + \lambda_v h$$
$$\lambda_v = 200$$
$$\lambda_o = 3.$$

The resulting mesh is shown in Fig. 4b. The grooves are better covered by the nodes, but the overall distribution of points is poor. A well-balanced distribution of points can be obtained by combining the two forms of adaption discussed above. Therefore use both the relative position of a node along the connecting lines and the shape of the interface. By using the set of parameters

$$w(x, z) = 1 + \lambda_v \exp(-\tau\mu)(\tfrac{1}{10} + h(1 - \mu))$$
$$\tau = 1$$
$$\lambda_v = 200$$
$$\lambda_o = 2,$$

TABLE III

Comparison of Aspect Ratios for Elements at a Highly Distorted Boundary

Mesh	Average aspect ratio
Fig. 4a	5.78
Fig. 4b	3.04
Fig. 4c	8.37

we obtain the mesh shown in Fig. 4c, which seems to have a good distribution of points, while placing more nodes in groove areas. A comparison of the average aspect ratios at the curved boundary for the three meshes in Fig. 4 is given in Table III.

Other aspects of the automatic mesh generation procedure are also important. Since the position of a node depends on the relative position of all other nodes, a change in any parameter affects the whole distribution of nodes. For example, increasing the number of nodes from a 50 × 20 mesh to a 100 × 20 mesh does not simply result in dividing each element in half, as would be the case in a regular fixed domain. In addition, one cannot assume that a set of parameters which seems ideal for one case, can simply be extrapolated to other cases. However, the functional form and parameters used to generate the mesh in Fig. 4c were found to be satisfactory for all the CVD cases we considered. Undoubtedly, one of the key aspects of the solution procedure is to choose mesh generation parameters which will ensure a good distribution of nodes for a wide variety of domain shapes.

5. ILLUSTRATIVE EXAMPLES

The methods discussed above are combined into a numerical solution procedure which can be used for studying the evolution of an interface in many free-boundary problems. Our focus is on solid film growth during CVD according to the model presented earlier. Specifically, examples are considered where diffusional limitations lead to a very nonuniform film structure, with a highly distorted gas–solid interface. For a specific set of deposition conditions and an arbitrary initial interface shape, the numerical solution procedure is as follows:

For a specific choice of resolution, i.e., the number of nodes along the interface and the number of quadrilateral elements inside the domain, an initial mesh is generated according to the optimization procedure described in the previous section. The field equation is then solved in steady state by the Galerkin finite element method with appropriate boundary conditions to determine the reactant concentration field in the gas phase. The concentration values at the interface, as well as the shape of the interface, enter into the solid phase mass balance, which is used to describe its movement. This equation contains terms with spatial derivatives expressed in terms of the parameter σ, which we take as the normalized arc-length. The interface is discretized on an equidistant grid in σ, and derivatives are expressed in a finite difference form. A fourth-order Runge–Kutta scheme is used for time integration. With each time step, the interface position changes and a new mesh has to be generated for the gas domain. The field equation is then solved again to determine the new concentration field, after which the position of the interface at the next time step can be determined. The concentration field and interface position are therefore determined in *successive* steps, and the solution scheme is explicit. This procedure is repeated until the desired deposition time is reached.

As an example, consider the growth of a solid film from an arbitrary interface with random roughness [24] under conditions leading to gas diffusional limitations. Typical values of the system parameters are used. These are:

Pe	4.62×10^{-5}
η	4.95×10^{-9}
Da	50.0
Γ (μm)	4.95×10^{-6}
l (μm)	250.0
βC_b	9.51×10^{-7}
ϕ	85.6
time (s)	3000

Evolution of the interface in time is shown in Fig. 5a and iso-concentration lines at the end of the run in Fig. 5b. The results show that a highly nonlinear interface shape develops and that the solution procedure is able to follow its development beyond the point where a single-valued representation becomes inappropriate. The iso-concentration lines in Fig. 5b show the depletion of reactant in grooves under diffusion-limited conditions, which explains their development and why the film becomes nonuniform. One should note that the gas domain extends well above the cutoff point in the graph and the reactant concentration at the top boundary is in fact $\mathscr{C} = 1.0$. The grid used for generating these results was 300 × 30, with strong adaption closer to the interface and larger elements far away from it, where the concentration field does not change rapidly.

Note that boundary conditions are periodic and the length of film or substrate considered can be chosen arbitrarily. However, one has to ensure that a sufficiently large part of the substrate is considered so that the boundaries do not affect development of the film structure. A rule of

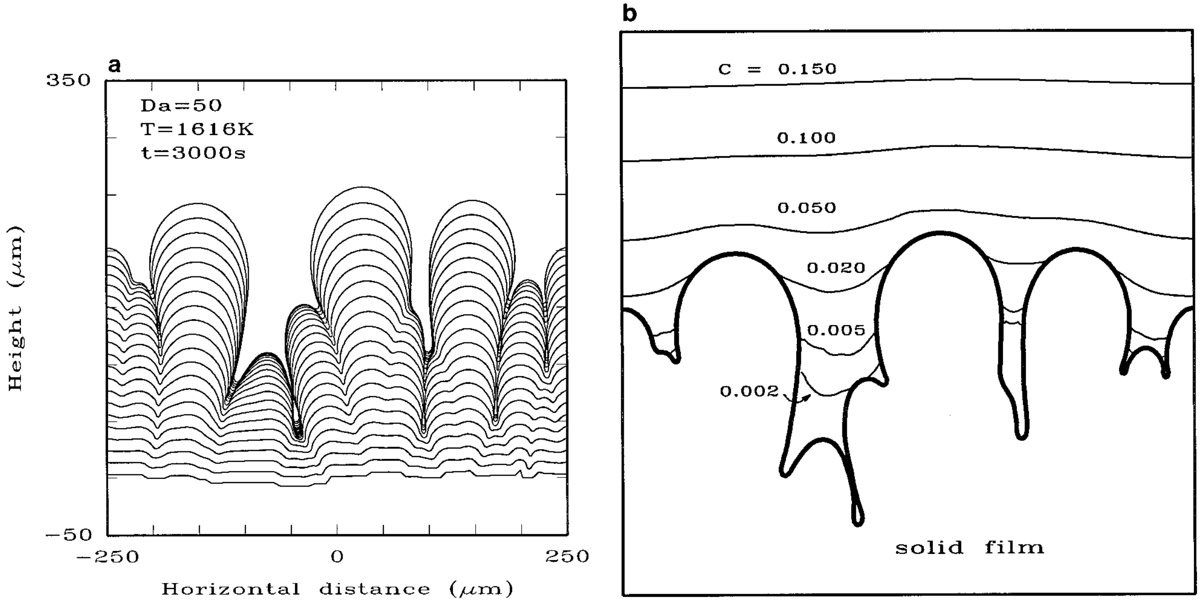

FIG. 5. Example of film growth during diffusion-limited CVD: (a) interface evolution; (b) iso-concentration lines at end of run.

thumb is to choose a length of substrate which is more than 10 times the wavelength of the *most* unstable perturbation, λ_m, as predicted by a linear stability analysis [24]. This ensures that several nodules can develop without being affected by computational boundaries. In the present example $\lambda_m = 35.8$ μm and the length of substrate considered is $l = 500$ μm. To further illustrate this point, consider film growth under the same conditions as before, but using two different lengths of computational domain. The first case is a substrate of 50 μm with random roughness, and the second case is a substrate of 500 μm with identical roughness. Deposition conditions are the same for both cases. The results are shown in Figs. 6a and 6b. Note that the film structures are completely different and the picture in Fig. 6b is not obtained by simply combining 10 lengths of the film in Fig. 6a. They have completely different characteristics. No grooves develop in the case with the 50 μm substrate, while they are very prominent in the 500 μm case. The results suggest that it is very important to consider a large enough domain, so that boundary conditions do not affect the development of the structure and lead to unphysical results.

Points on the interface move normal to their previous position, which means the spacing of points will decrease in some areas and increase in others. In the present study, the number of nodes along the interface is kept constant, and to avoid problems with uneven resolution, the nodes along the interface are rearranged in an equidistant distribution each time before the domain solution is obtained. This will lead to an overall smooth solution, but it may cause details in small grooves to be lost. This problem can be avoided by using a large number of points from the start, which may be computationally very expensive. Alternatively, the number of points can be increased as the total arc-length increases, thus keeping the resolution essentially constant. Usually in an explicit solution scheme, extremely small time steps are required to ensure solution stability. The critical time step depends on the size of space steps and the order of derivatives in the governing equation, and they can be calculated a priori in a fixed grid problem with constant coefficients [25]. In the present case of CVD growth, it should be noted that spatial derivatives up to fourth order are present, which is apparent if the interface mass balance is written in the conventional coordinate system. This is similar to what was encountered in some flame propagation problems [26] and means that the relation between the maximum allowable time step Δt and grid spacing Δx follows $\Delta t \propto (\Delta x)^4$. Therefore, there is a strong incentive from a computational standpoint to limit the number of nodes along the interface. An interesting observation is that the parametrization which leads to space steps in arc-length (thus $\Delta\sigma$), instead of the horizontal coordinate (Δx), seemed to introduce a damping effect on solution stability. We found that the maximum allowable time step for a specific number of spatial points could be increased by about 50% if the interface description is changed from the conventional single-valued $\mathcal{H}(x)$ to the parametric form $\mathcal{H}(\sigma)$, with σ as the arc-length. This may be due to the frequent rearrangement of points along the interface in the latter case.

When comparing computational time requirements of different parts of the solution procedure, one finds that the mesh generation and Galerkin solution steps are by far the most time-consuming. The question arises whether

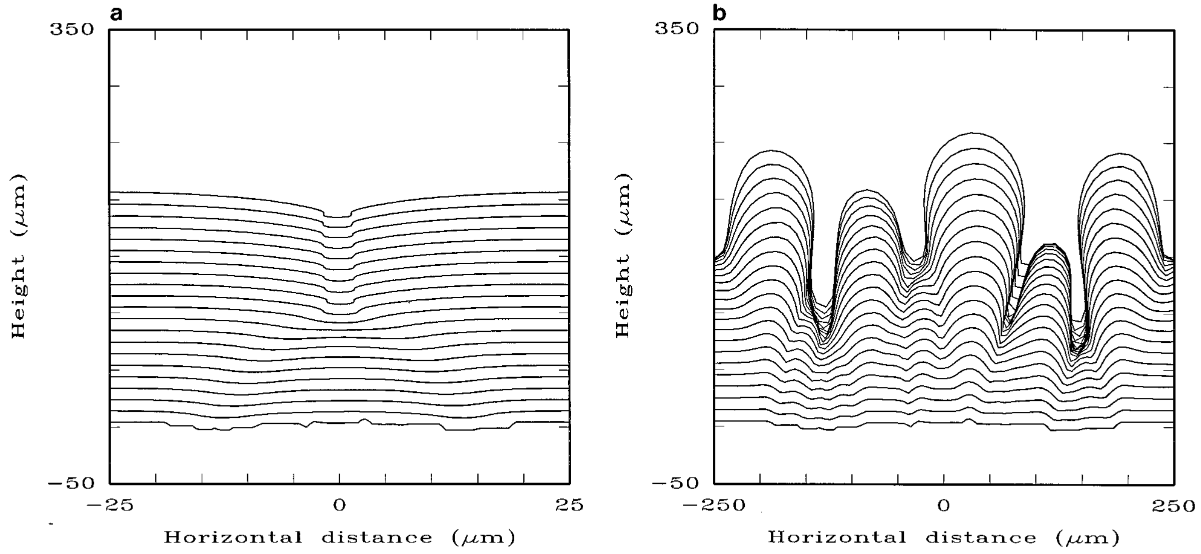

FIG. 6. Effect of length of domain considered on the development of film structure: (a) 50 μm; (b) 500 μm, with all other conditions identical.

it is really necessary to perform these steps at each time step, since the concentration solution depends on the interface shape, which changes slowly in time. To study this effect, consider an initial interface consisting of a combination of sine curves with different wavelengths, with deposition occurring under the same conditions as in Fig. 5. Three cases with different frequency of updating the domain solution are shown in Fig. 7. In the first case, the concentration solution is obtained at every time step, as shown in Fig. 7a. If the concentration is updated every five time steps, the results in Fig. 7b are obtained. The difference in results is hardly noticeable, while a significant savings in computational time is achieved. If, however, the concentration solution is obtained only every 40 time steps, the results become more inaccurate and are significantly different, as shown in Fig. 7c. The most obvious difference is that the width of the grooves has decreased. Table IV summarizes the differences in computational time and accuracy. These results indicate that it is not necessary to update the concentration solution at every time step and that significant savings in computational time can be achieved without sacrificing accuracy. However, care has to be exercised in doing so and a compromise between accuracy and computational expense has to be found.

The number of nodes along the interface will also affect the accuracy of solution. Clearly, the number of points should be large enough to accurately account for the development of a highly nonlinear interface structure. A useful parameter in choosing the number of points is the wavelength of *most* unstable perturbation, λ_m, which gives a good indication of the scale of nodules developing into a characteristic film structure. There should be several points (>10) for each distance of λ_m along the interface. Consider an example where deposition takes place under the same conditions as before, but from a different initial interface with random roughness and using three cases with different numbers of nodes: 100, 200, and 400. The results are shown in Fig. 8. They indicate that there is remarkably little difference in the qualitative picture obtained in the three cases. The number of nodules or fingers which develop is essentially the same in each case, although they are better defined when more points are used. There is an 18% difference in results between the 100 point and 400 point case, but only 5.1% difference between the 200 and 400 point case. The most significant difference between the graphs is in the height of the fingers which develop, and they are largest in the case with 400 points. Another difference lies in the development of grooves, which are deeper and better defined in the case with most points.

The results suggest, that a very good indication of the interface structure can be obtained by using a relatively coarse mesh, which will lead to significant savings in computational time. What does seem to be important is the choice of computational domain or *length* of interface,

TABLE IV

Effect of Frequency of Updating Concentration Solution on Computational Requirements and Accuracy

Update frequency	Normalized computational time	Percentage difference in results
Every time step	1.0	Base case
Every 5 time steps	0.21	2.7
Every 40 time steps	0.04	34

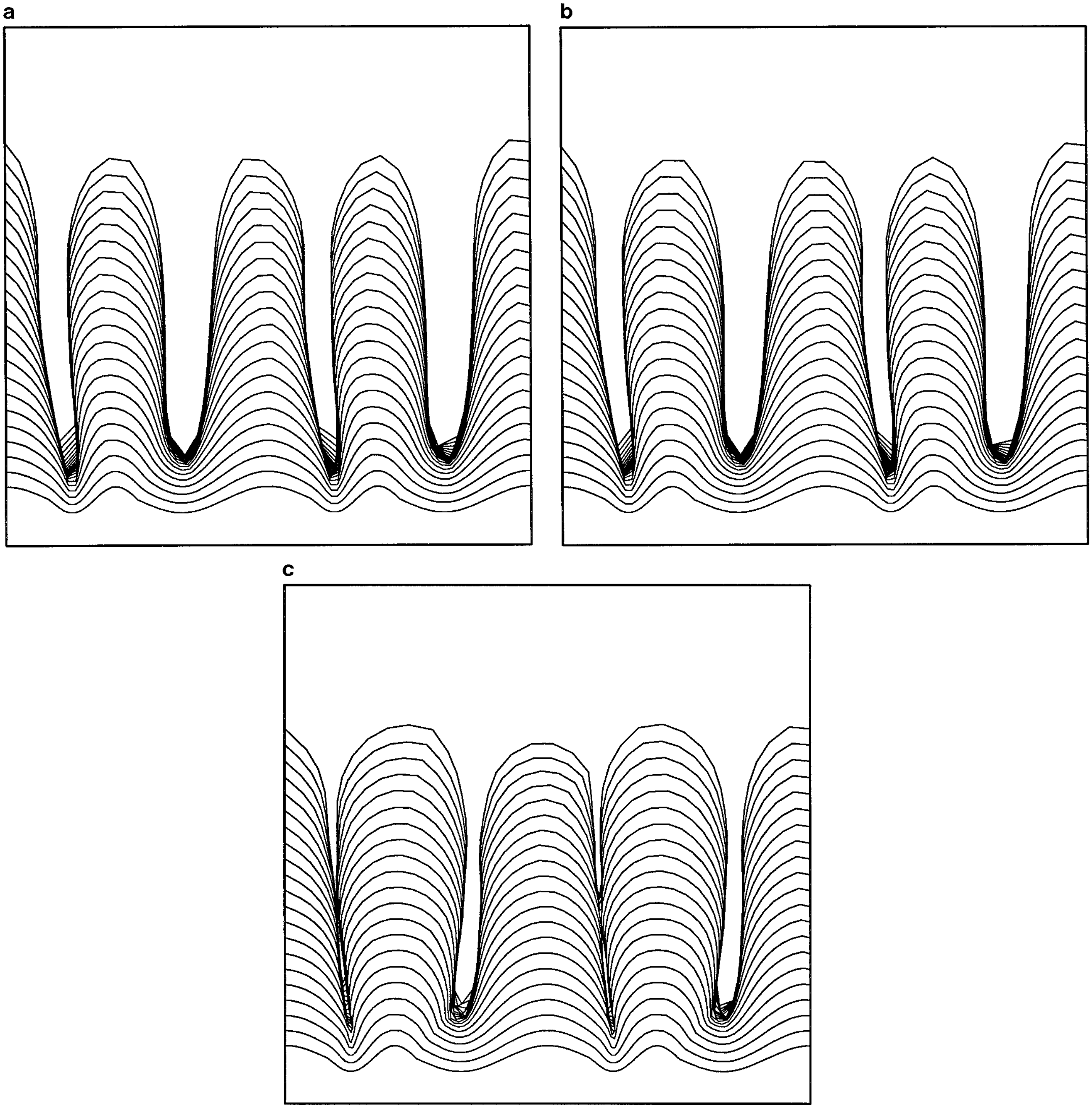

FIG. 7. Effect of updating concentration solution less frequently on the evolution of the interface: (a) update every time step; (b) update every five time steps; (c) update every 40 time steps.

which must be large enough to allow a representative structure to develop. Once this is achieved, refinement of the mesh will improve the results *quantitively,* but they should not change them *qualitatively.*

6. SUMMARY AND CONCLUSIONS

In this study we describe a numerical procedure for solution of moving boundary problems with highly distorted interface shapes. To illustrate application of this method, we study the growth of solid films during chemical vapor deposition (CVD), where a highly nonlinear gas–solid interface develops under conditions of diffusion-limited deposition. Special attention is given to two aspects of the solution procedure. The first is expression of the interface shape in parametric form, which allows the evolution of virtually any shape. The second is an elliptic mesh generation procedure, which determines the location of

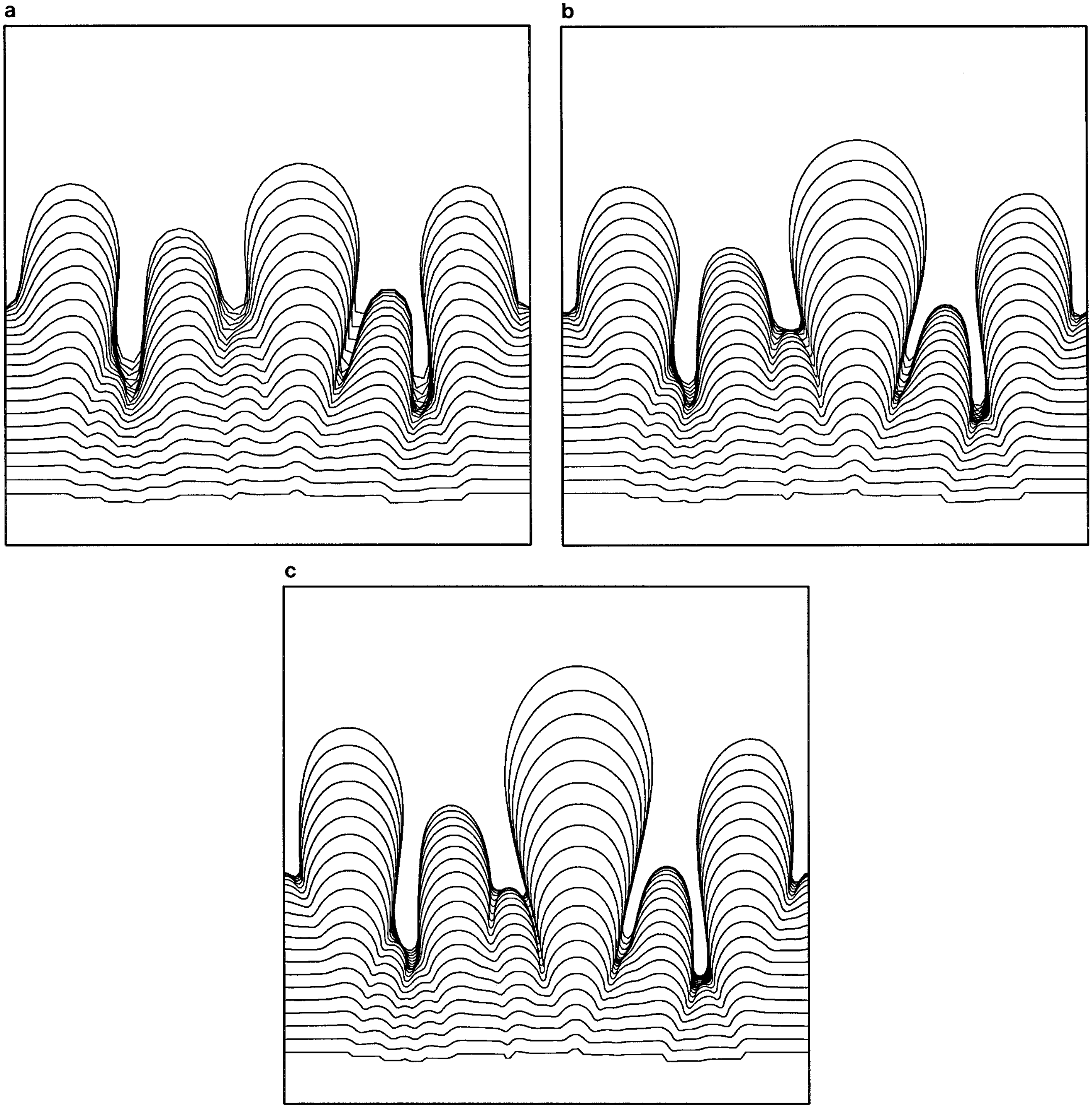

FIG. 8. Effect of number of points on development of film structure: (a) 100 points; (b) 200 points; (c) 400 points.

nodes inside an arbitrary domain by minimizing an objective function accounting for important mesh properties.

Simulation results show that the relatively simple solution procedure can be used to follow interface evolution beyond the point where a single-valued representation of the interface fails. It is shown that mesh generation parameters can be used to control the placement of nodes inside the changing domain. It was expected that the number of nodes placed along the interface would be critical to the attainment of realistic results and a representative film structure, but this is not the case. What seems more important is the *length* of interface or boundary chosen for simulation. Care should be taken to ensure that this is large enough to ensure that results are not affected by boundary conditions.

APPENDIX

Notation

CVD	chemical vapor deposition
C_b	concentration of reactant in bulk flow, mole/m^3
$\mathscr{C}$	reactant concentration
D_f	gas diffusion coefficient, m^2/s
D_s	surface diffusion coefficient, m^2/s
D_s^*	modified surface diffusivity, $D_s \Omega \alpha$, m^3/s
Da	Damköhler number, defined as $K\delta/D_f$
E	objective function to be minimized
$\mathscr{H}$	position of interface
I_{s}	functional of grid smoothness
I_{o}	functional of grid orthogonality
I_{v}	functional of weighted volume variation
J	Jacobian
l	period of interface considered, *m*
$\mathbf{n}$	vector normal to surface
N_{AV}	Avogadro's number
Pe	Peclet number, $V_0\Gamma/D_f$
s	arc-length along interface
t	time
$\mathbf{t}$	vector tangential to surface
v	rate growth
w	weight function
x	horizontal coordinate
z	vertical coordinate

Greek Symbols

α	number of molecules per unit surface area, 1/m^2
β	molar volume, ΩN_{AV}, m^3/mole
$\gamma(\mathscr{H}, \mathscr{C})$	film growth term
Γ	Capillary length, m
δ	boundary layer thickness, m
η	defined as Γ/δ
κ	surface curvature
λ_{m}	wavelength of *most* unstable perturbation, μm
λ_{o}, λ_{v}	scalar weights
μ	mesh generation parameter
σ	parameter used for interface representation
ϕ	dimensionless effective surface diffusivity, $D_s^*/D_f\Gamma$,
$\chi(\mathscr{H})$	surface diffusion term
Ω	molecular volume, m^3

ACKNOWLEDGMENT

We gratefully acknowledge the contribution Jan Degreve, who developed a code for adaptive mesh generation used in the calculation of combustion fronts. We modified and used his code in our examples.

REFERENCES

1. M. S. Longuet-Higgins and E. D. Cokelet, *Proc. R. Soc. London A* **350,** 1 (1976).
2. G. M. Homsy, *Annu. Rev. Fluid Mech.* **19,** 271 (1987).
3. D. Bensimon *et al., Rev. Modern Phys.* **58**(4), 977 (1986).
4. S.-C. Huang and M. E. Glicksman, *Acta Metallurg.* **29,** 701 (1981).
5. G. I. Sivashinsky, *Acta Astronaut.* **4,** 1177 (1977).
6. J. J. Thiart *et al., Combust. Sci. Technol.* **82,** 185 (1992).
7. G. S. Bales and A. Zangwill, *J. Vac. Sci. Technol. A* **9**(1), 145 (1991).
8. H. J. Viljoen, J. J. Thiart, and V. Hlavacek, *AIChE J.* **40**(6), 1032 (1994).
9. L. I. Rubinstein, *The Stefan Problem,* Transl. Math. Monographs, Vol **27** (Amer. Math. Soc., Providence, RI, 1971).
10. R. V. Southwell, *Relaxation Methods in Theoretical Physics* (Oxford Univ. Press, London, 1949), p. 201.
11. H. J. Oh, S. W. Rhee, and I. S. Kang, *J. Electrochem. Soc.* **139**(6), 1714 (1992).
12. K. N. Christodoulou and L. E. Scriven, *J. Comput. Phys.* **99,** 39 (1992).
13. K. Tsiveriotis and R. A. Brown, *Int. J. Numer. Methods Fluids* **14,** 981 (1992).
14. H. M. Ettouney and R. A. Brown, *J. Comput. Phys.* **49,** 118 (1983).
15. J. J. Thiart, V. Hlavacek, and H. J. Viljoen, *Chem. Eng. Sci.* **50,** 3493 (1995).
16. W. W. Mullins, *J. Appl. Phys.* **28**(3), 333 (1957).
17. Q. S. Bhatia and V. Hlavacek, *Chem. Eng. Commun.* **22,** 287 (1983).
18. M. Kubicek, *ACM Trans. Math. Software* **2**(1), 98 (1976).
19. R. Seydel and V. Hlavacek, *Chem. Eng. Sci.* **42**(6), 1281 (1987).
20. J. Degreve *et al., Comput. Chem. Eng.* **11**(6), 749 (1987).
21. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation: Foundations and Applications* (Elsevier Science, New York, 1985).
22. J. U. Brackbill and J. S. Saltzman, *J. Comput. Phys.* **46,** 342 (1982).
23. J. Degreve, Dissertation, SUNY at Buffalo, 1990 (unpublished).
24. J. J. Thiart and V. Hlavacek, *AIChE J.,* **41,** 1926 (1995).
25. B. A. Finlayson, *Nonlinear Analysis in Chemical Engineering* (McGraw-Hill, New York, 1980).
26. D. M. Michelson and G. I. Sivashinsky, *Acta Astronaut.* **4,** 1207 (1977).