JOURNAL OF COMPUTATIONAL PHYSICS 110, 373–398 (1994)

Numerical Simulation of the Ion Etching Process

S. J. SHERWIN AND G. E. KARNIADAKIS

Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, New Jersey 08540

AND

S. A. ORSZAG

Program in Applied and Computational Mathematics, Princeton University, Princeton, New Jersey 08540

Received September 24, 1992; revised August 6, 1993

Numerical techniques traditionally used in the simulation of compressible fluid dynamics are applied to the ion etching process. This process is governed by a non-linear hyperbolic conservation law describing the evolution of the local slope of a surface. The hyperbolic nature of the equation allows discontinuities of slope to develop which are seen numerically and experimentally as cusps in the surface of the etched material. These discontinuities are analogous to shocks in fluid dynamics. Initially, an *essentially non-oscillatory* (ENO) algorithm is used to simulate the evolution of a single homogeneous material with fixed boundaries and known flux function. The algorithm is then extended to simulate the evolution of two different homogeneous materials which is more representative of a typical etching configuration. The two materials are assumed to be separated by an interface of known form. The additional mathematical and physical reasoning to describe the two-material configuration is presented from which a new algorithm is developed. This algorithm requires the hyperbolic conservation law to be solved on a moving mesh since the interface between the materials is numerically treated as a moving boundary. The nature of the two-material problem is such that shocks and expansion waves can develop at this interface and thus special numerical treatment of the moving boundary is required; this is achieved by using a lower order approximation in this localised region. Finally, a more realistic method to calculate the flux function is adopted which changes the nature of the governing equation since the flux function becomes dependent on the geometry of the surface as well as the local slope. The algorithm is extended to include this flux calculation which allows the numerical simulation of the physically observed phenomena such as *RIE lag* and *undercutting*.

1. INTRODUCTION

In an age of ever-increasing miniaturisation the need for tools to help understand the microchip fabrication process becomes more important. The ultimate aim of this branch of research is to numerically simulate this fabrication process, thereby giving design engineers an effective tool to optimise chip design as well as to allow further very large scale integration (VSLI). Ion etching is part of the fabrication process and is generally preferred over other types of etching because of its highly anisotropic nature.

The ion etching model we have adopted assumes a known energy and angular distribution of energetic particles incident upon a surface. We also assume that all etched materials have a constant number density (i.e., a fixed number of particles per unit volume), and that we have knowledge about the energy and angular dependence of the yield per incident particle. That is to say, we have information describing the rate at which a surface dissolves for a specified ion bombardment at a given angle and energy. This model was proposed by Jurgensen and Shaqfeh [1] and is believed to account for some of the major factors in ion etching. However, this model assumes that all particles either stick or react with the surface at their initial point of contact. This means that there are no glancing angle reflections. In reality this may not be the case, especially at large angles of incidence. We also neglect any other mechanisms such as ion steering by microelectric fields. Preliminary research of this mechanism has been conducted by Arnold and Sawin [2]. We do not specifically account for any deposition of polymers in the plasma or redeposition of etched material. Nevertheless, this may be dealt with in the yield function if redeposition can be modeled as a function of angle and energy distribution.

The objective of this paper is to present a consistent algorithm to numerically simulate this ion etching model for configurations which have practical applications in the ion etching industry. Previous numerical simulation have been performed only recently. Three different approaches have been considered: Ross [3] adopted a Lax Wendroff finite-difference scheme with artificial viscosity to deal with a single material evolution. Jurgensen and Shaqfeh [4] adopted a method of characteristics approach which solves an ordinary differential equation along the characteristic paths. Another simulation has been done by Pelka *et al.* [5]

0021-9991/94 $6.00

using the string algorithm. This algorithm represents the surface as a string of points and then moves these points according to their local etch rate. However, problems occur at corners and regions where the surface varies rapidly. To overcome this a special algorithm is used which is detrimental to the generality of the scheme. The method of characteristics has the disadvantage that it is only valid in smooth regions. When characteristics intersect the solution must be treated locally as a Riemann problem. The formulation used by Ross has the most general applications. However, instead of using the Lax Wendroff finite-difference scheme which requires the use of artificial viscosity to solve hyperbolic problems we have adopted an *essentially non-oscillatory* (ENO) approach to solving the slope equation.

This paper is organised as follows: In Section 2 we present the mathematical model describing the ion etching process in a single domain as well as the treatment of the interface between two materials. In Section 3 the numerical solution to this model is presented for a single material using the ENO scheme. Some validation and convergence results are shown. In Section 4 the numerical algorithm is extended to deal with a two-material configuration and results using this algorithm are shown in Section 5. Finally in Section 6 the flux function is modified to deal with shadowing effects as proposed by Jurgensen and Shaqfeh [4, 1]. This extension is numerically implemented, and results demonstrating the physically observed phenomena are presented.

2. MATHEMATICAL MODEL

2.1. Governing Equation

Jurgensen and Shaqfeh [1] obtained the following evolution equation for the configuration shown in Fig. 1,

$$\frac{\partial G}{\partial t} + u\frac{\partial G}{\partial x} - v = 0, \tag{1}$$

where $G(\mathbf{X}, t)$ is the local height of the surface and u and v are the local surface velocities in the x and y directions, respectively. Differentiating this equation with respect to x we obtain

$$p_t + (u \cdot p - v)_x = 0, \tag{2}$$

where

$$p = \partial G/\partial x.$$

We will call Eq. (2) the *slope equation*. The slope equation describes the evolution of the slope along a one-dimensional interface which can be integrated to reconstruct a two-dimensional surface. If u and v are purely functions of the slope p this equation is a hyperbolic conservation law, where the flux function is given by

$$f(p) = up - v. \tag{3}$$

This is the case in regions where the incident particles have linear trajectories which are limited only by the local surface slope. Nevertheless, if the trajectories are obscured by a non-local point the flux function will depend on the slope p and the limiting trajectory angle B. In practice, there are two angles (B_{upp}, B_{low}) which define an integration region that determines the local surface velocities (u, v).

If the surface is raised, isolated, and convex (Fig. 2a) then it is said to be *locally self-shadowed* and the angle B will be purely dependent upon the local slope p. On the other hand, if the surface is *shadowed* due to the surface geometry (Fig. 2b) then the angle B is dependent on the local coordinates as well as those of the shadowing point (x_{sh}, y_{sh}). This means that u and v are functions of $(p, x_{sh}, y_{sh}, x, y)$ and the governing slope equation is

$$p_t + f(B, p)_x = 0, \tag{4}$$

where $B = B(p)$ in a locally self-shadowed region and $B = B(p, x_{sh}, y_{sh}, x, y)$ in a shadowed region.

Two typical flux functions for the unshadowed slope equation are plotted in Fig. 3. The first flux function is similar to that found experimentally by Lee [6] and used by Ross [3]. The probability of a collision resulting in an atom acquiring enough momentum directed away from the surface to leave the material increases as the angle of incidence increases. However, as the incident angles of particles become very large the ion flux drops off and the probability of purely elastic reflection of the ions is increased. This causes the turning points in the flux function at incident

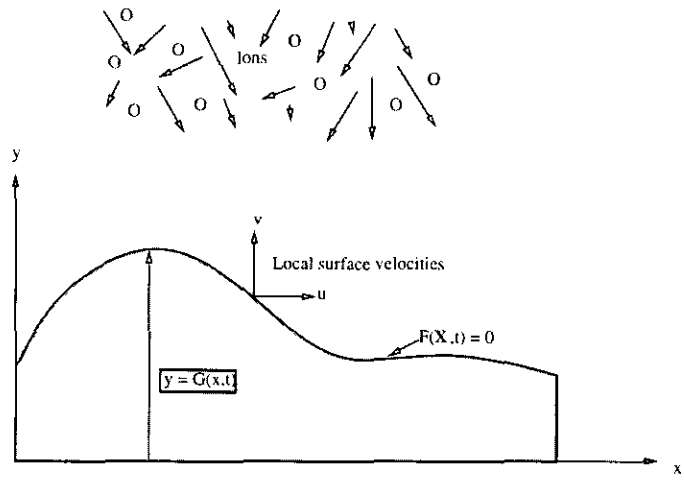


FIG. 1. Etching configuration for mathematical model.

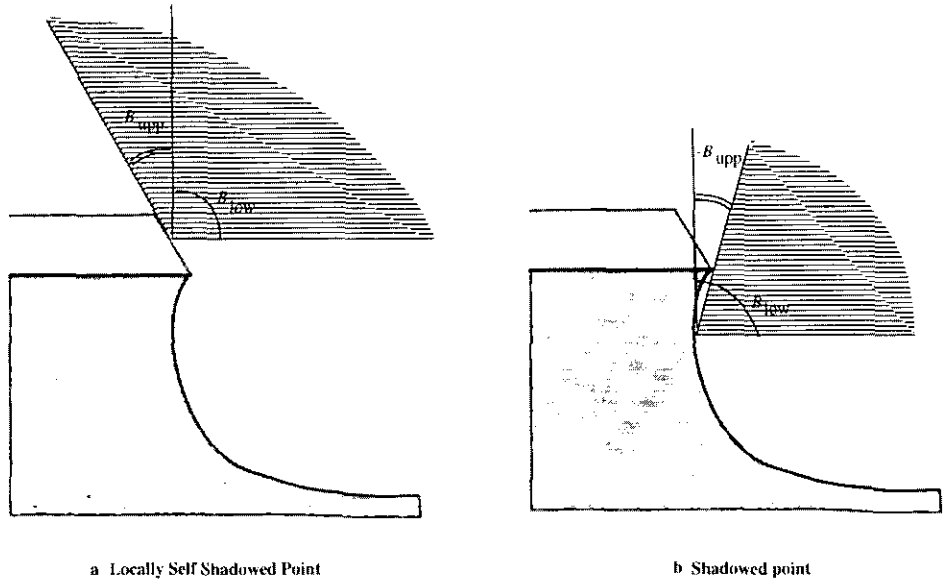


FIG. 2. Shadowing regions due to surface geometry.

angles of about 40°–60° (i.e., $p \approx \pm 1$). The second flux function is representative of a flux function calculated numerically by assuming that the yield is independent of the angular distribution of incident particles [4]. Therefore, etching is proportional to the energy and flux of particles integrated over an appropriate domain (in this case an unshadowed domain). It can be appreciated that since there will be particles incident, and therefore etching, on the surface at all slope values the flux function tends to infinity at large slope values.

2.2. Moving Boundary Conditions between Materials

In extending the model to deal with two materials we have two slope equations, one for each material. The configuration is shown in Fig. 4, where we have an upper and lower material separated by an interface described by a known function $g(x)$.

The boundary condition between these materials has been studied by Ross [7]. After some manipulation of the slope and evolution equations for each material, (1), (2), we can derive a Rankine Hugoniot condition for the moving boundary of the form

$$\frac{f^u(p_b^u(t))}{p_b^u(t) - g'(x_b(t))} = \frac{dx_b(t)}{dt} = \frac{f^l(p_b^l(t))}{p_b^l(t) - g'(x_b(t))}, \tag{5}$$

where p_b^u, p_b^l are the boundary values of the slope in the upper and lower material, respectively, and $x_b(t)$ is the x location of the moving boundary.

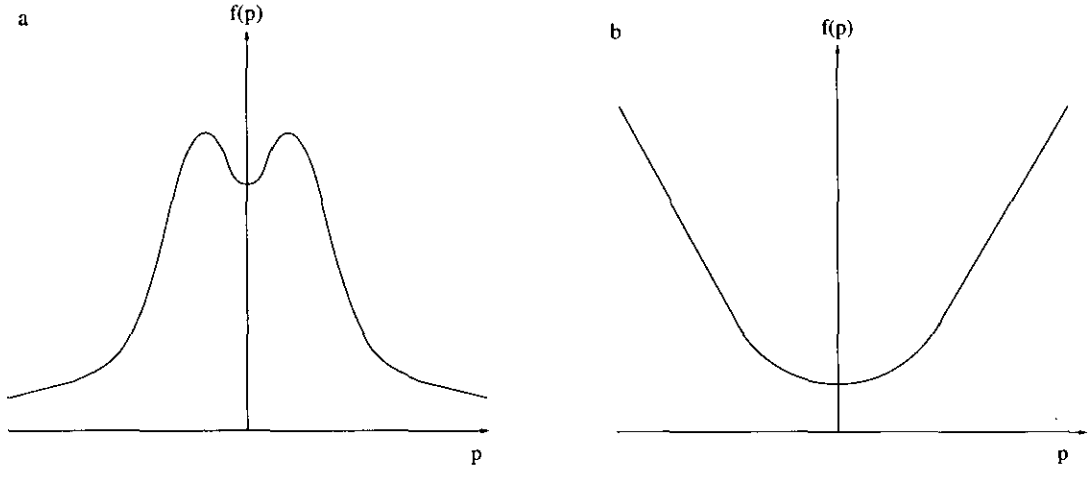


FIG. 3. Flux functions for slope equation: (a) angle and energy dependent flux; (b) purely energy-dependent flux.

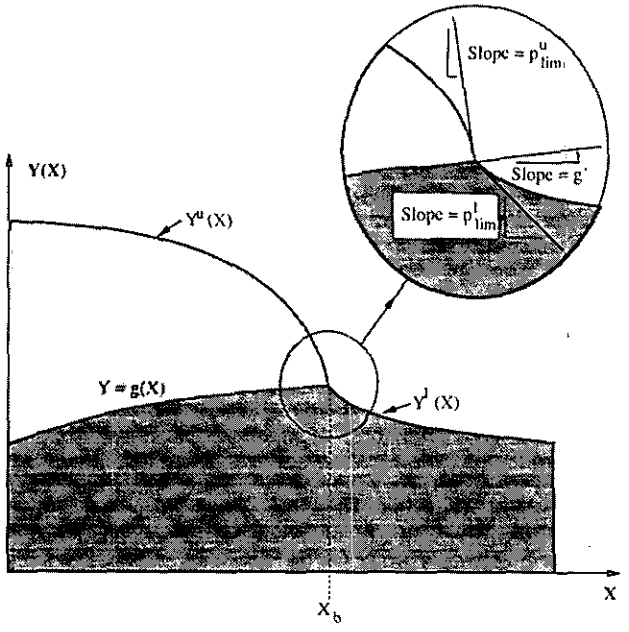


FIG. 4. Two material configuration.

However, the correct boundary value of the slope in the upper surface p_b^u may not be the limit value of the slope $p_{\lim} = \lim_{x \to x_b} p^u(x, t)$. The reason for this is due to some physical restrictions which must be imposed at the moving boundary. These are that the upper surface boundary condition is independent of the lower surface evolution and that the lower surface boundary condition is purely dependent on the evolution of the upper surface. This is quite rational since the lower surface cannot etch until the upper surface has been eroded. Mathematically, this means that the upper surface must be an outflow boundary condition and the lower surface must be an inflow boundary condition. Ross [7] derived the following conditions to determine the upper surface boundary condition under this physical rationale,

$$\frac{f(p_b^u(t))}{p_b^u(t) - g'(x_b(t))}(p - g'(x_b(t))) \geqslant f(p) \quad \text{for all} \quad p < p_{\lim}, \tag{6}$$

if the speed of the moving boundary, dx_b/dt, is positive. Similarly,

$$\frac{f(p_b^u(t))}{p_b^u(t) - g'(x_b(t))}(p - g'(x_b(t))) \geqslant f(p) \quad \text{for all} \quad p > p_{\lim}, \tag{7}$$

if the speed of the moving boundary, dx_b/dt, is negative. The sign of the moving boundary direction is given by the material configuration.

Having determined the correct upper surface boundary condition p_b^u and the associated boundary speed dx_b/dt we need to determine the lower surface boundary condition p_b^l. Due to the nature of the flux function there may not be a unique root to the lower surface Rankine Hugoniot condition given by the second equation in (5). The physical reasoning that the boundary should be an inflow determines a unique root and the mathematical inequality which must be imposed to determine this root is

$$|f(p_b^l) - f(p)| < \left| \frac{dx_b}{dt}(p_b^l - p) \right| \tag{8}$$

for all p between $p_{\lim}$ and p_b. Here $p_{\lim}$ denotes the limit value of the slope in the lower surface at the boundary $p_{\lim} = \lim_{x \to x_b} p^l(x, t)$. This is the similar to the entropy condition for the lower material.

3. SINGLE MATERIAL FORMULATION

The ENO scheme adopted here (Fig. 5) is based on a paper by Harten *et al.* [8]; here we have adopted a Runge–Kutta time-stepping scheme rather than a Taylor expansion scheme proposed in that paper. This is because in general we do not have perfect knowledge of the flux function and to use the time stepping scheme proposed by Harten *et al.* we need to be able to calculate higher derivatives of the flux function.

We solve the slope equation (2) in the conservative form,

$$\bar{p}_j^{n+1} = \bar{p}_j^n + \lambda(\bar{f}_{j+1/2} - \bar{f}_{j-1/2}), \tag{9}$$

where $\lambda = \Delta t/h$ (assuming a uniform mesh and constant time step) and $\bar{p}_j^n$ is the *cell-averaged* value of the slope at time level $t = t^n$ in cell j defined by

$$\bar{p}_j^n = \bar{p}(x_j, t^n) = \frac{1}{h_j} \int_{x^{j-1/2}}^{x^{j+1/2}} p(x, t^n)\, dx, \quad \text{where} \quad h_j = x_{j+1/2} - x_{j-1/2}. \tag{10}$$

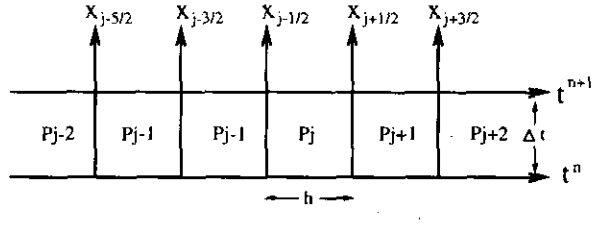


FIG. 5. Nomenclature for the ENO scheme.

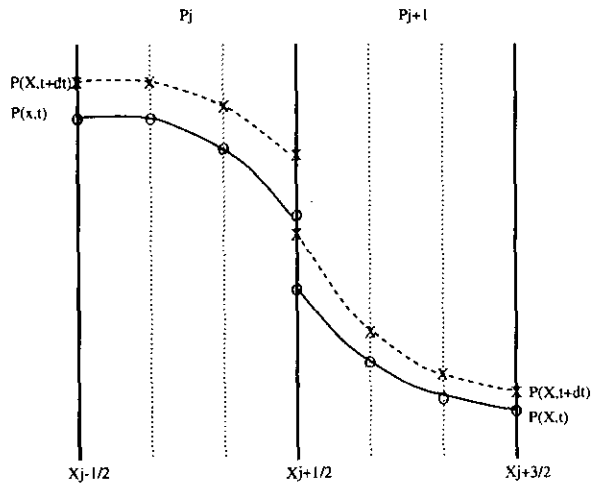


FIG. 6. Discretization required to maintain spatial accuracy in the Runge–Kutta time integration.

$\bar{f}_{j+1/2}$ is the time-averaged flux at the location $x = x_{j+1/2}$ defined by

$$\bar{f}_{j+1/2} = \bar{f}(p(x_{j+1/2}, t)) = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(p(x_{j+1/2}, t))\, dt,$$

$$\text{where} \quad \Delta t = t^{n+1} - t^n. \tag{11}$$

To numerically evaluate the conservative form of the conservation law the scheme can be described in three distinct steps: (i) *reconstruction*, (ii) *solution in the small*, and (iii) *cell averaging*.

In the reconstruction step, we reconstruct a piecewise polynomial approximation of the variable $p(x, t^n)$ from the cell averages $\bar{p}_j^n$ using *reconstruction via primitive function*. Next, we propagate the solution in the small in each cell to determine an approximation to $p(x, t)$ at t^{n+1} from our solution at t^n. Finally, we close the loop by updating the cell averages using Eq. (9). This involves evaluating the flux value at the cell boundaries which is done by using an approximate Riemann solver. In the present scheme we have used the modified Roe solver as proposed by Harten *et al.* [8] and an exact solver which is a discrete approximation to the exact Riemann solution.

As mentioned previously the solution in the small is evaluated by using a Runge–Kutta method as proposed by Shu and Osher [9]. We are primarily interested in evaluating the solution only at the cell boundaries so we can calculate the time-averaged flux. Nevertheless, to maintain high order accuracy in this spatial operator we have to interpolate the solution at r points in each cell. We cannot use our reconstruction stencil as the slope values may not be continuous at the mesh points. Here r denotes the number of points used in the reconstruction stencil. Figure 6 demonstrates this point. We initially have a piecewise cubic solution as shown by the continuous line marked $p(x, t)$. In order to evaluate the spatial operator in each cell we have to propagate the solution at the boundaries as well as the two intermediate points (shown by the dotted lines). These four points allow us to calculate the spatial derivative to an order of accuracy consistent with the cubic representation of the solution. Finally, after evaluating the Runge–Kutta algorithm we end up with a cubic approximation to the solution at time $t + \Delta t$ as shown by the dashed line. We note that the solution remains discontinuous at the cell boundary $x_{j+1/2}$.

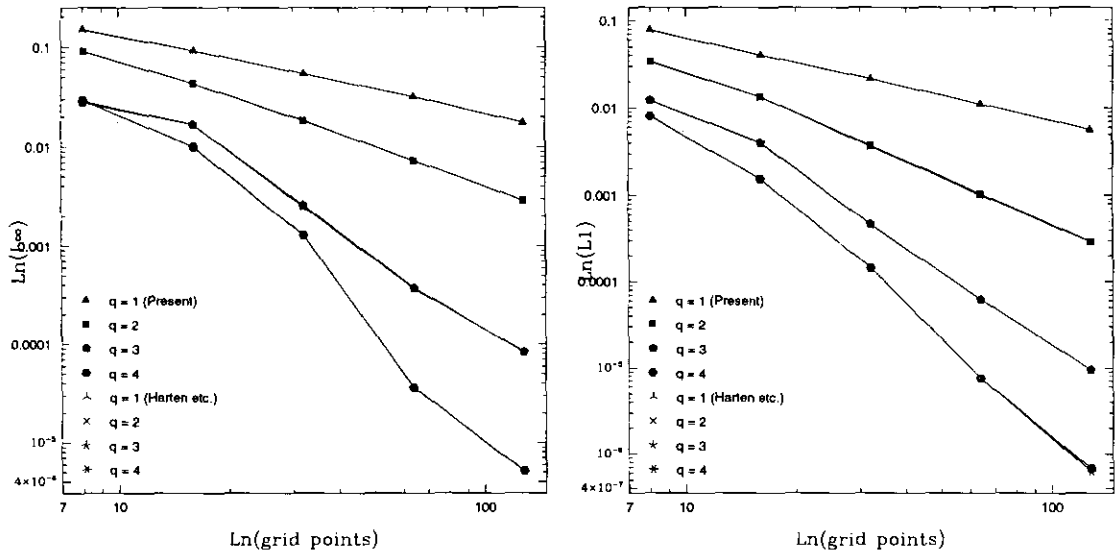


FIG. 7. Convergence errors for test case using Burgers equation with exact inflow boundary conditions. Comparison of current test with results of Harten *et al.* [8].

3.1. Validation and Convergence Tests

As means of validation and to check the convergence of the algorithm, using the Runge–Kutta time stepping scheme, test runs with the inviscid Burgers equation were made. These test cases were taken from the paper by Harten *et al.* [8].

We use the initial condition:

$$u(x, 0) = 1.0 + \tfrac{1}{2}\sin(\pi x)$$

and a CFL condition of 0.6. The solutions were computed to a final time $t_{\text{final}} = 0.3$ time units. The computations were made for $-1 \leqslant x \leqslant 1$, $t \geqslant 0$ using Roe's approximate Riemann solver. The form of the solution for this problem is a shock developing when $t = 2/\pi$. The final time is well below this occurrence and therefore the exact solution can be calculated using a Newton–Raphson iteration on the characteristic equation. The results from Harten *et al.* [8] used a time stepping scheme which is of the same order as the spatial discretisation. The results from the present scheme use a third-order Runge–Kutta algorithm, as proposed by Shu and Osher [9], in all calculations regardless of spatial accuracy.

A typical test case is plotted in Fig. 7 which shows L_∞ and L_1 errors plotted versus the number of grid points. Each set of lines corresponds to a different spatial accuracy q which means that a stencil of $r = q + 1$ was used. The ENO scheme is equivalent to Godunov's scheme for a first-order calculation and we note that exactly the same results are achieved in this case. All these results are identical to the original ENO scheme presented by Harten *et al.* [8].

The second test case used by Harten *et al.* [8] used a non-convex flux function. This test was the Riemann problem,

$$u_t + f(u)_x = 0, \qquad u(x, 0) = \begin{cases} u_L, & x < 0 \\ u_R, & x > 0, \end{cases}$$

where the flux function $f(u)$ was taken to be

$$f(u) = \tfrac{1}{4}(u^2 - 1)(u^2 - 4).$$

In this case we used the exact Riemann solver. Two test cases were run with the initial conditions,

$$u_1(x, 0) = \begin{cases} 2, & x < 0 \\ -2, & x > 0 \end{cases}$$

and

$$u_2(x, 0) = \begin{cases} -3, & x < 0 \\ 3, & x > 0. \end{cases}$$

The first diagram in Fig. 8 shows the relevant upper convex envelope for the correct entropy solution to the Riemann problem, test case 1. Similarly, the second diagram in Fig. 8 shows the lower convex envelope relevant to the second Riemann problem, test 2. From this we note that the solution to the first problem will involve two shocks separated by an expansion fan whilst the second problem involves two expansion regions separated by a stationary shock.

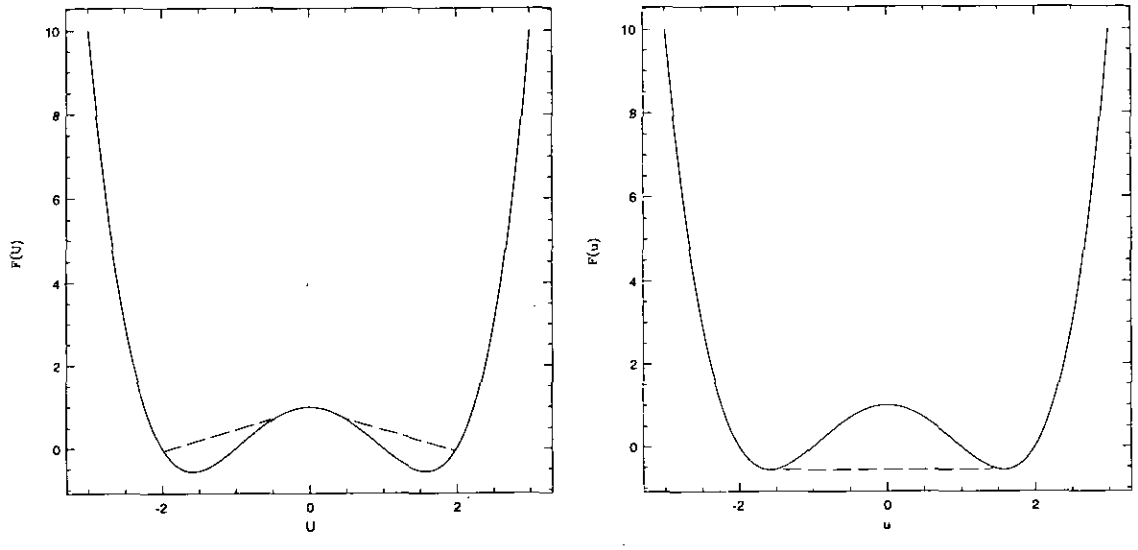


FIG. 8. Quartic flux function for Riemann tests. Dashed lines show the upper and lower convex envelopes.

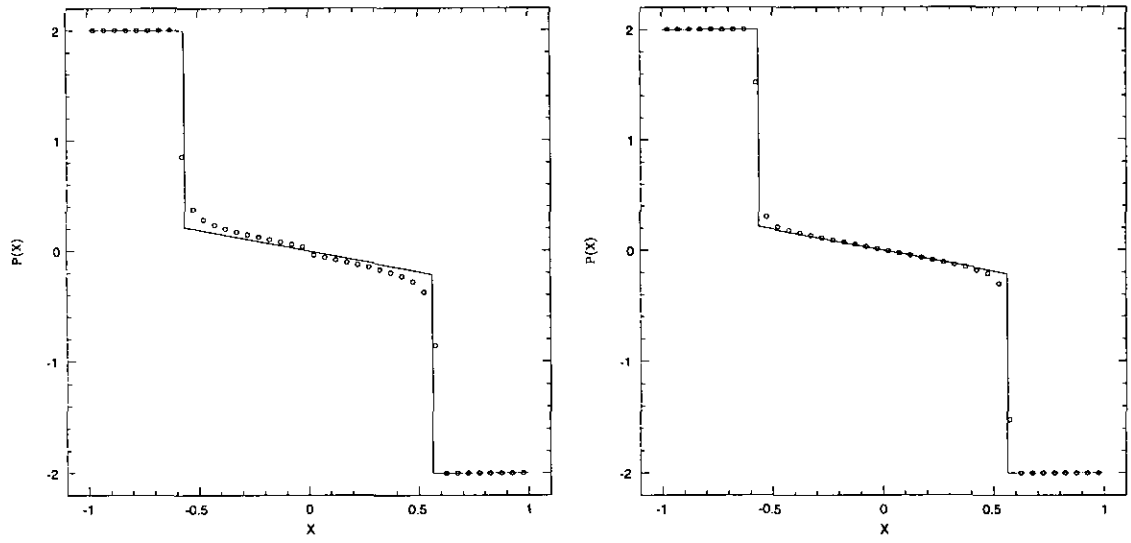


FIG. 9. Solution to Riemann test case 1 with a first- and third-order accurate scheme. Solid line shows exact solution.

The test cases were both calculated with a CFL condition of 0.8 in the domain $-1 < x < 1$ with 40 equispaced mesh points using an exact Riemann solver. The first test case was calculated for 80 time steps and the second for 20 time steps, both using a third-order Runge–Kutta scheme. The solutions to test 1 and test 2 can be seen in Figs. 9 and 10 respectively for first- and third-order accurate schemes. The exact solution is shown by the continuous line and the numerical results are shown by the circles. In the first test case we see that we capture the location of the shock and expansion wave satisfactorily. In the second test case we see that the stationary shock is perfectly resolved. Harten *et al.* [8] noted that the modified Roe solver for this case had the correct structure although the propagation speed was incorrect. Nevertheless, when the test was refined the correct entropy solution was obtained.

3.2. Single Material Ion Etching Results

The following results use a flux function given by Ross [3],

$$f(p) = \begin{cases} 0.0601219\cos(\theta) & -0.0192006\cos(3\theta) \\ +0.0006436\cos(5\theta) & -0.0002343\cos(7\theta) \\ -0.0004133\cos(9\theta) & +0.0004361\cos(11\theta) \\ -0.0002564\cos(13\theta) & +0.0002366\cos(15\theta), \end{cases} \tag{12}$$

where $\theta = \arctan(p)$.

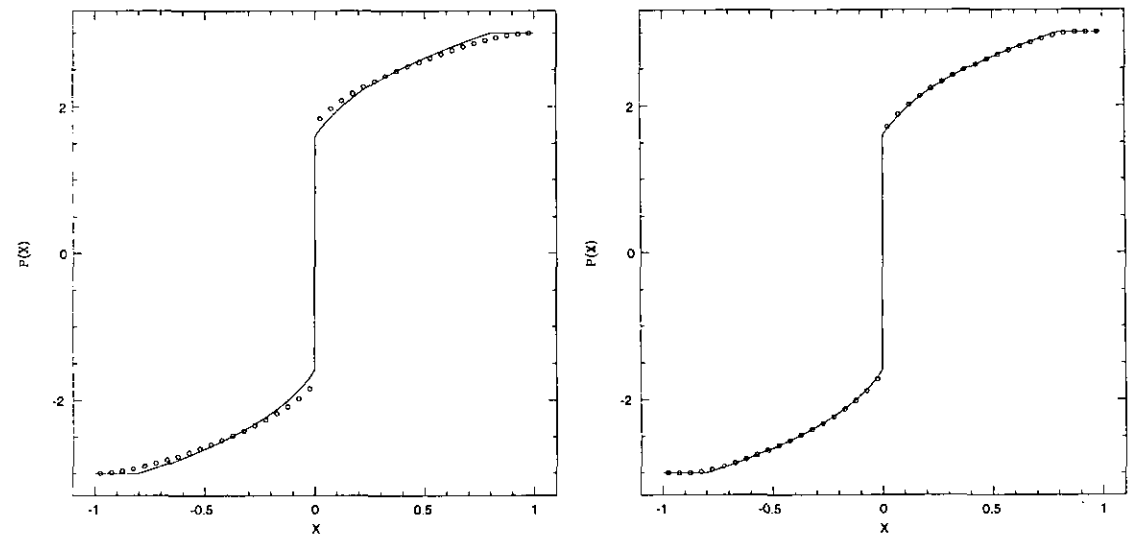


FIG. 10. Solution to Riemann test case 2 with a first- and third-order accurate scheme. Solid line shows exact solution.

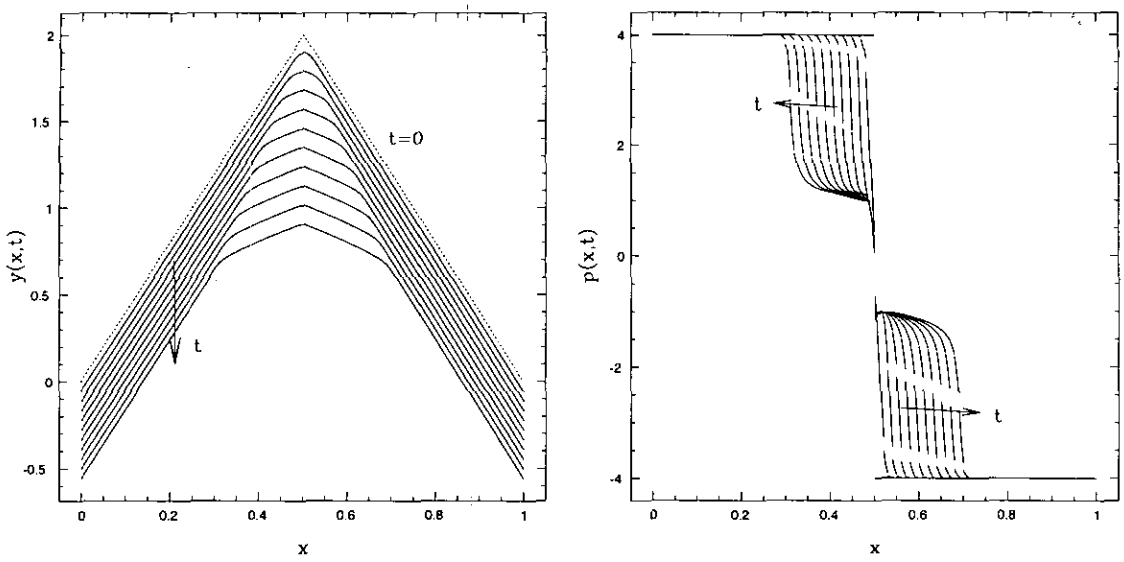


FIG. 11. Numerical simulation of Riemann problem in (x, y) and (x, p) planes. Each line corresponds to solution after a period of two time units.

In these runs we only consider profiles that have constant slope values at the boundaries of the computational domain. If an inflow boundary condition is required we use the appropriate constant slope values. The calculations for these runs are done in the (x, p) plane and to recover the (x, y) plane we effectively integrate $p(x, t)$ by summing the cell averages. However, we do require an additional integration constant which is evaluated in a region of constant slope using Eq. (1). If p is a constant then we can integrate Eq. (1) to obtain

$$y(x_p, t) = y(x_p, 0) - f(p) \cdot t,$$

where x_p is an x location which has constant slope.

The first result corresponds to the solution of a Riemann problem. Here we start with an initial condition of $p^- = 4$, $p^+ = -4$ which corresponds to the upper line in the first plot in Fig. 11. This figure shows the evolution of the material at time intervals of two units. In the (x, p) plane we note the development of three shocks shown by the near vertical regions. These are separated by two expansion regions. We also note the piecewise polynomial representation of the solution; the gaps between the solution occur at the cell boundaries. In the (x, y) plane the shocks are seen as cusps in the surface forming facet corners whilst the expansion regions can be seen as regions of smooth curvature between these corners. The evolution of the surface for $t > 0$ is in the negative y direction.

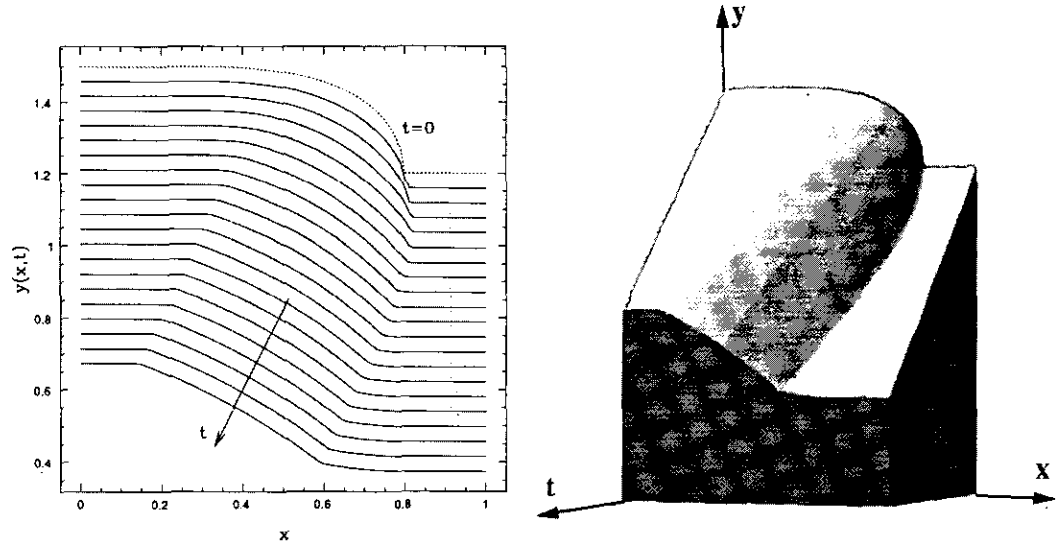


FIG. 12. Surface evolution of a test case used by Ross (see text). Each line corresponds to a time interval of one unit. The initial profile contains only one shock (cusp in the surface) which moves to the right and then changes direction whilst a second shock develops on the left-hand side.

The second test case is a demonstration run used by Ross [3]. The above flux function is used to calculate the evolution of a profile which has an initial condition of two horizontal lines connected by the curve:

$$y(x) = 1.5 + 0.65(x - 0.3) + 0.92(\sqrt{0.8008 - x} - \sqrt{0.5008})$$

$$\text{for} \quad 0.3 < x < 0.8.$$

The results in the (x, y) plane can be seen in Fig. 12. Again the initial profile is given by the uppermost line and evolution for $t > 0$ is in the negative y direction. The lines correspond to the evolution of the surface at intervals of one time unit. The initial surface contains one shock at $x = 0.8$. This shock initially propagates towards the right but is intersected by a second weak shock which reverses the direction forming a single shock moving towards the left. Another shock also develops from the initially smooth region at $x = 0.3$. This shock also propagates towards the left.

4. TWO MATERIALS FORMULATION

The next step is to include a second material into the simulation. As described in Section 2.2, given a configuration with a known interface slope, $g'(x_b)$ and the limit values of the slopes in the upper and lower materials, p^u_{lim} and p^l_{lim}, we can determine the moving speed, dx_b/dt, and the boundary conditions for the slopes, p^u_b and p^l_b. The problem here is that the limit slope value p_{lim} may not necessarily be the same as the boundary slope value p_b. To connect the limit value p_{lim} to the boundary value p_b a shock or expansion region may develop. For this to happen in our numerical scheme the moving boundary must be a mesh point so that the correct boundary value can be imposed. To achieve this we introduce a linear stretching in the x coordinate. The motivation for such a transformation came from a paper on the numerical solution to a Stefan problem by Rønquist and Patera [10].

4.1. Co-ordinate Stretching

The co-ordinate stretching is of the form

$$\hat{x} = \frac{(x_b(t_0) - c)}{(x_b(t) - c)}(x - c) + c, \tag{13}$$

where $x_b(t)$ is the location of the moving boundary at time t, t_0 is the initial time, and c is taken to be a fixed datum point in the x plane, for example, the left boundary if the right boundary is moving. If we let $\tau = t$ we can perform a change of variables in the following fashion: Denoting the value of the slope in terms of the new variables, $(\hat{x}, \tau)$ by $\hat{p}$, we find that

$$d\hat{p} = \left.\frac{\partial \hat{p}}{\partial \hat{x}}\right|_\tau d\hat{x} + \left.\frac{\partial \hat{p}}{\partial \tau}\right|_{\hat{x}} d\tau \tag{14a}$$

$$d\hat{x} = \left.\frac{\partial \hat{x}}{\partial x}\right|_t + \left.\frac{\partial \hat{x}}{\partial t}\right|_x dt \tag{14b}$$

$$d\tau = dt \tag{14c}$$

and substituting Eqs. (14b) and (14c) into Eq. (14a) we obtain

$$d\hat{p} = \left.\frac{\partial \hat{p}}{\partial \hat{x}}\right|_\tau \beta\, dx + \left(\left.\frac{\partial \hat{p}}{\partial \tau}\right|_{\hat{x}} + v \left.\frac{\partial \hat{p}}{\partial \hat{x}}\right|_\tau\right) dt, \tag{15}$$

where

$$\beta = \left.\frac{\partial \hat{x}}{\partial x}\right|_\tau = \frac{x_b(t_0) - c}{x_b(t) - c}$$

$$v = \left.\frac{\partial \hat{x}}{\partial t}\right|_x = -\frac{(x_b(t_0) - c)}{(x_b(t) - c)^2} \cdot x \cdot \frac{dx_b(t)}{dt}$$

$$= -\frac{(\hat{x} - c)}{(x_b(t) - c)} \frac{dx_b(t)}{dt}.$$

A change of slope in the original co-ordinates is given by

$$dp = \left.\frac{\partial p}{\partial x}\right|_t dx + \left.\frac{\partial p}{\partial t}\right|_x dt; \tag{16}$$

combining Eqs. (15) and (16) we find

$$\left.\frac{\partial p}{\partial x}\right|_t = \beta \left.\frac{\partial \hat{p}}{\partial \hat{x}}\right|_\tau \tag{17a}$$

$$\left.\frac{\partial p}{\partial t}\right|_x = v \left.\frac{\partial \hat{p}}{\partial \hat{x}}\right|_\tau + \left.\frac{\partial \hat{p}}{\partial \tau}\right|_{\hat{x}}. \tag{17b}$$

Therefore we can rewrite the slope equation

$$p_t + f'(p)\, p_x = 0$$

by substituting in Eqs. (17a) and (17b) as

$$\hat{p}_\tau + v\hat{p}_{\hat{x}} + f'(\hat{p})\, \beta \hat{p}_{\hat{x}} = 0. \tag{18}$$

This equation is similar to the untransformed equation, except that it has an extra convective term, $v\hat{p}_{\hat{x}}$. We see that the characteristics of the transformed equation are given by the solution to the equation

$$\frac{d\hat{x}}{dt} = v + \beta f'(\hat{p}). \tag{19}$$

Since

$$\frac{d\hat{x}}{dt} = \beta \frac{dx}{dt} + v$$

we see that by combining these two equations we recover the original characteristic equation,

$$\frac{dx}{dt} = f'(p).$$

Noting that v is linearly dependent on $\hat{x}$ and β is independent of $\hat{x}$ we can rewrite Eq. (18) as

$$\hat{p}_\tau + (v\hat{p} + \beta f(\hat{p}))_{\hat{x}} = v_{\hat{x}}\,\hat{p}. \tag{20}$$

It is possible to use a more general transformation of the form

$$\hat{x} = \left(\frac{x_b^r(t_0) - x_b^l(t_0)}{x_b^r(t) - x_b^l(t)}\right)(x - x_b^l(t)) + x_b^l(t_0),$$

where x_b^l and x_b^r are the left and right moving boundaries, respectively.

In this case,

$$\beta = \frac{x_b^r(t_0) - x_b^l(t_0)}{x_b^r(t) - x_b^l(t)}$$

$$v = -\left[\frac{\hat{x} - x_b^l(t_0)}{x_b^r(t) - x_b^l(t)}\right]\left[\frac{dx_b^r}{dt} - \frac{dx_b^l}{dt}\right] - \beta \frac{dx_b^l}{dt}.$$

This form is needed when solving for the two-material configuration, since the moving boundary will be on the left in one domain and on the right in the other. However, the external boundaries in each domain are usually fixed, which eliminates some of the terms in the expressions above. For example, if $x_b^l(t) = c$ the transformation above reduces to Eq. (13). It should also be appreciated that if the computational domain is translated at the speed of the moving boundary (i.e., $dx_b^r/dt = dx_b^l/dt$) then $v_{\hat{x}} = 0$. This is to be expected since in this case there is no grid divergence and the co-ordinate mapping is pure translation.

4.2. Time Splitting

We now wish to solve a hyperbolic conservation law with a source term on the right-hand side of Eq. (20). To do this we choose to split the equation in the following fashion: Starting with the equation,

$$\hat{p}_t + G_{\hat{x}} = v_{\hat{x}}\,\hat{p},$$

where $G = G(\hat{p}, \hat{x}, t) = v(\hat{x}, x_b(t), x_b'(t))\,\hat{p} + \beta(x_b(t))\,f(\hat{p})$ and integrating over a time interval we obtain

$$\hat{p}^{n+1} - \hat{p}^n + \int_{t^n}^{t^{n+1}} G_{\hat{x}}\,dt = \int_{t^n}^{t^{n+1}} v_{\hat{x}}\,\hat{p}\,dt. \tag{21}$$

Note that the nomenclature here is the same as the original formulation for a single material, Fig. 5, except that the grid is now in $\hat{x}(x, t)$ coordinates instead of x. Therefore the slope p is denoted as $\hat{p}$. Splitting this equation to deal with the flux term and the source term individually reduces Eq. (21) to

$$q = \hat{p}^n - \int_{t^n}^{t^{n+1}} G_{\hat{x}}\,dt \tag{22}$$

$$\hat{p}^{n+1} = q + \int_{t^n}^{t^{n+1}} v_{\hat{x}}\,\hat{p}\,dt. \tag{23}$$

If we integrate over a cell in space from $\hat{x}_{j-1/2}$ to $\hat{x}_{j+1/2}$ we recover an equation similar to the conservative form of the slope equation (9),

$$\int_{\hat{x}_{j-1/2}}^{\hat{x}_{j+1/2}} q_j\,d\hat{x} = \int_{\hat{x}_{j-1/2}}^{\hat{x}_{j+1/2}} \hat{p}_j^n\,d\hat{x} - \int_{t^n}^{t^{n+1}} [G(\hat{p}, \hat{x}_{j+1/2}, t) - G(\hat{p}, \hat{x}_{j-1/2}, t)]\,dt. \tag{24}$$

Defining

$$\bar{\hat{p}}_j^n = \frac{1}{h}\int_{\hat{x}_{j-1/2}}^{\hat{x}_{j+1/2}} \hat{p}(\hat{x}, t^n)\,dx$$

and

$$\bar{G}_{j+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} G(\hat{p}, \hat{x}_{j+1/2}, t)\,dt,$$

Eq. (24) becomes

$$\begin{aligned} \bar{q}_j &= \bar{\hat{p}}_j^n - \lambda[\bar{G}_{j+1/2} - \bar{G}_{j-1/2}] \\ \lambda &= \Delta t/h, \end{aligned} \tag{25}$$

where

$$\bar{q}_j = \frac{1}{h}\int_{\hat{x}_{j-1/2}}^{\hat{x}_{j+1/2}} q\,dx.$$

We solve this equation in a similar manner to that described in Section 3. We reconstruct the solution as a piecewise polynomial approximation from the cell averages $\bar{\hat{p}}$. Then we propagate the solution in the small using a Runge–Kutta scheme. In this case we have to solve the

transformed equation (20), which involves the extra convective term. Finally we update the cell average $\bar{\hat{p}}_j^n$ to determine the pseudo-cell average $\bar{q}_j$. However, the major difference is that the flux function G is no longer a function of the slope alone which means that the characteristics, given by Eq. (19), are no longer straight lines. This makes the solution of the Riemann problem, to determine the correct flux value at the cell boundaries, more complicated. Nevertheless, we note that the solution to the Riemann problem,

$$\hat{p}_t + G'(\hat{p}, \hat{x}_0, t_0)\,\hat{p}_{\hat{x}} = 0 \qquad \text{with} \quad \hat{p}(\hat{x}, 0) = \begin{cases} \hat{p}^-, & x < 0 \\ \hat{p}^+, & x > 0, \end{cases}$$

where

$$G'(\hat{p}, \hat{x}_0, t_0) = \left.\frac{\partial(G(\hat{p}, \hat{x}, t))}{\partial \hat{p}}\right|_{\hat{x}=\hat{x}_0,\, x_b = x_b(t_0),\, x_b' = x_b'(t_0)}$$

and $\hat{x}_0$, t_0 are the fixed location and time, has characteristics given by

$$\frac{d\hat{x}}{dt} = G'(\hat{p}, \hat{x}_0, t_0)$$
$$= v(\hat{x}_0, x_b(t_0), x_b'(t_0)) + \beta(x_b(t_0))\, f'(p).$$

The characteristics of this problem are the same as the characteristics of the transformed problem given by Eq. (19) in the limit $\hat{x} \to \hat{x}_0$ and $t \to t_0$. Since we are only interested in solving the Riemann problem at a known $\hat{x}$ location and time t we use the solution to an approximate Riemann solver with a flux $f(p) = G(\hat{p}, \hat{x}_0, t_0)$ rather than the exact Riemann problem for this situation where the characteristics are given by Eq. (19).

Having determined the pseudo cell average q_j we now need to complete the time step by evaluating the second part of the time splitting, Eq. (23). First we note that the coefficient of the source term $v_{\hat{x}}$ is independent of $\hat{x}$. This is seen easily if we evaluate this term:

$$v_{\hat{x}}(x_b(t)) = \frac{-dx_b(t)/dt}{x_b(t) - c}.$$

Therefore, we can integrate Eq. (23) over a cell in space from $\hat{x}_{j-1/2}$ to $\hat{x}_{j+1/2}$ and divide by the grid spacing h to obtain

$$\frac{1}{h}\int_{\hat{x}_{j-1/2}}^{\hat{x}_{j+1/2}} \hat{p}^{n+1} \cdot d\hat{x} = \frac{1}{h}\int_{\hat{x}_{j-1/2}}^{\hat{x}_{j+1/2}} q \cdot d\hat{x} + \frac{1}{h}\int_{\hat{x}_{j-1/2}}^{\hat{x}_{j+1/2}}\int_{t^n}^{t^{n+1}} v_{\hat{x}}\,\hat{p}(\hat{x}, t)\, dt \cdot d\hat{x}.$$

From the definitions of the cell averages we obtain

$$\bar{\hat{p}}_j^{n+1} = \bar{q}_j + \frac{1}{h}\int_{\hat{x}_{j-1/2}}^{\hat{x}_{j+1/2}}\int_{t^n}^{t^{n+1}} v_{\hat{x}}\,\hat{p}(\hat{x}, t)\, dt\, d\hat{x}.$$

Now if we make a trapezoidal approximation to the time integral of the form

$$\int_{t^n}^{t^{n+1}} v_{\hat{x}}\,\hat{p}(\hat{x}, t)\, dt = \frac{\Delta t}{2}\left[v(x_b(t^{n+1})) \cdot p(\hat{x}, t^{n+1}) + v(x_b(t^n)) \cdot p(\hat{x}, t^n)\right]$$

and since $v_{\hat{x}}$ is independent of $\hat{x}$, we can approximate Eq. (23) as

$$\bar{\hat{p}}_j^{n+1} = \bar{q}_j + \frac{\Delta t}{2h}\left[v_{\hat{x}}(x_b(t^{n+1}))\,\bar{\hat{p}}_j^{n+1} + v_{\hat{x}}(x_b t^n)\,\bar{\hat{p}}_j^n\right].$$

This can be re-arranged to give a cell average correction to the pseudo cell average $\bar{q}_j$ of the form

$$\bar{\hat{p}}_j^{n+1} = \left[1 - \frac{\Delta t}{2h} v_{\hat{x}}(x_b(t^{n+1}))\right]^{-1} \times \left[\bar{q}_j + \frac{\Delta t}{2h} v_{\hat{x}}(x_b(t^n))\,\bar{\hat{p}}_j^n\right]. \tag{26}$$

4.3. Scheme Outline and Interpretation

In summary the scheme for a two-material configuration has the following outline:

- Determine the correct upper surface boundary condition p_b^u using either inequality (6) or (7) and the limit value of the slope p_{lim}^u.
- Propagate the cells adjacent to the moving boundary to determine $x_b(t)$ and $dx_b(t)/dt$ using the Rankine Hugoniot condition (5). We can then determine the lower boundary condition p_b^l using the limit slope p_{lim}^l and $dx_b(t)/dt$ and inequality (8).
- Reconstruct the slope values $\hat{p}(x, t^n)$ as a piecewise polynomial approximation from the cell values $\bar{\hat{p}}_j^n$.
- Propagate piecewise polynomials forward in time using a Runge–Kutta scheme to solve Eq. (20).
- Solve the Riemann problem using the flux $f(p) = G(\hat{p}, \hat{x}, x_b)|_{\hat{x}=\hat{x}_0,\, x_b = x_{b0}}$.
- Approximate the time averaged flux $\bar{G}$ using numerical quadrature and calculate the pseudo-cell average $\bar{q}_j$ using Eq. (25).
- Correct the pseudo-cell average $\bar{q}_j$ using Eq. (26) to evaluate $\bar{\hat{p}}_j^{n+1}$.

- Update the x values using

$$x(t+\Delta t)=\frac{(x_b(t^{n+1})-c)}{(x_b(t^n)-c)}(\hat{x}-c)+c.$$

In determining the upper boundary condition p_b^u, the process of evaluating the inequality (6) or (7) is minimized if the turning points of the function $f(p)/(p-g'(x_b(t)))$ are determined. This is because the correct boundary value will either be the slope value at a turning point of this function or the limit slope value. This does assume that the interface slope $g'(x_b(t))$ is a constant which is not too limiting an assumption for many etching configurations. The lower boundary condition is determined by initially making a coarse search to find the correct root as defined by the inequality (8) and then using a faster converging algorithm, like a Newton–Raphson iteration, to determine the root exactly. Finally, it should be noted that, although the mesh size h is changing at each time level we do not have to recalculate the derivative coefficients, used in the reconstruction stage, since they can be stored as multiples of h for an equispaced mesh.

Some insight into the effect of performing this co-ordinate stretching can be acquired by considering the test Riemann problem, shown in Fig. 11. We consider the initial condition, where the left-hand state is larger than the right, $p^- > p^+$. For this case we construct the convex envelope, which helps determine the correct entropy satisfying solution, for an untransformed and transformed flux function. The convex envelopes can be seen in Fig. 13: the first figure shows the envelope for the untransformed flux $f(p)$ and the second figure shows the envelope for the transformed flux $h(p)=G(p, x_0, t_0)$ (assuming a moving boundary with negative speed). In the first case the shape of the convex envelope would imply that the solution is a stationary shock between two shocks moving with equal and opposite speed separated by expansion regions. However, the solution for the transformed flux would be two shocks moving with positive speed and a third with negative speed, separated by expansion regions. Nevertheless, we see that the originally stationary shock is now moving with a speed which is equal and opposite to the mesh speed $(-v)$ at a given x location. The right-hand shock, shown by the line passing through $(p^+, f(p^+))$, is faster in the transformed plane since it will have to move against the mesh to maintain its original position. Similarly, the left-hand shock moves slower in the transformed plane since it is moving in the same direction as the mesh. Therefore, it can be appreciated that the same entropy satisfying solution will be achieved in each case; this can be seen by the numerical solution to this Riemann case in Sections 3.2 and 5.

4.4. Remeshing

Remeshing becomes an issue since we need to maintain a reasonable CFL condition,

$$\frac{|v_{\max}|\,\Delta t}{h}<1.$$

However, $|v_{\max}|$ is not easily defined here since the moving boundary speed dx_b/dt can be infinite; this occurs as the slope of the upper surface $p_{\lim}^u$ approaches the interface slope $g'(x_b(t))$. Nevertheless, if we assume that the moving

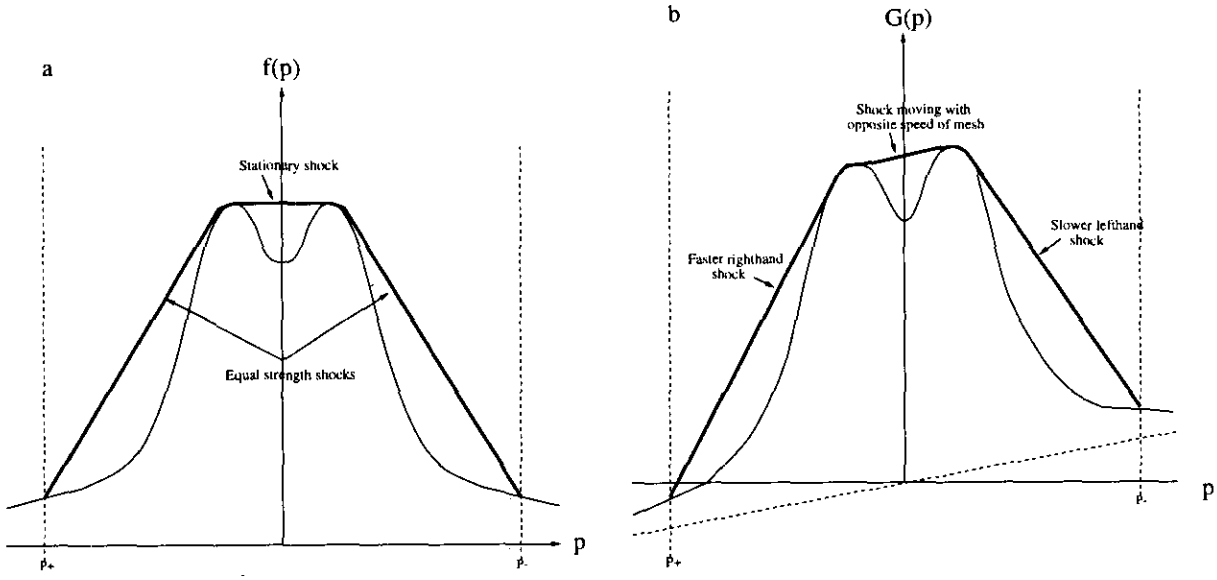


FIG. 13. Upper convex envelopes corresponding to a test case Riemann problem: (a) shows envelope for untransformed flux; (b) shows envelope for transformed flux.

boundary speed is bounded by $|dx_b/dt| < (dx_b/dt)_{\max}$, we can define $|v_{\max}|$ as the fastest characteristic speed:

$$|v_{\max}| = v_{\max} + \beta_{\max} \max_p (f'^u(p), f'^l(p))$$

$$v_{\max} = \beta_{\max}(dx_b/dt)_{\max}$$

$$\beta_{\max} = \frac{x_b(t_0) - c}{x_b(t_0) - \Delta t(dx_b/dt)_{\max} - c}$$

$$\approx 1 + O\left(\frac{\Delta t(dx_b/dt)_{\max}}{x_b(t_0) - c}\right),$$

for a time interval from t_0 to $t_0 + \Delta t$. In general, a working value of $\beta_{\max} \approx 1$, so $|v_{\max}|$ can be considered as the maximum absolute moving boundary speed plus the maximum absolute gradient of the upper and lower flux functions.

It can be appreciated that if we allow the mesh spacing to be reduced indefinitely then as h tends to zero, Δt must tend to zero in order to maintain the CFL condition. To overcome this we define a minimum mesh size $h_{\min}$ and remesh the grid whenever $h \leqslant h_{\min}$. The major requirement for the remeshing is that it be conservative. This will be satisfied if

$$\sum_{j=1}^{j=N} \bar{p}_j^{\text{old}} = \sum_{i=0}^{i=K} \bar{p}_i^{\text{new}}.$$

Typically $K = N - 1$ since we usually only decrease the domain by one cell at a time. Harten *et al.* [8] have shown that the reconstruction is conservative in the sense that

$$\frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} R(x, \hat{p})_j \, dx = \hat{p}_j,$$

where $R(x, \hat{p})_j$ denotes the piecewise polynomial approximation to $p(x, t^n)$ from the cell averages $\bar{p}^n$ in cell j.

One consistent way to remesh is shown in Fig. 14. Here we evaluate the new cell average $\bar{p}_i^{\text{new}}$ by adding the areas under the reconstructed curve of $p(x, t)$ and then dividing by the new mesh spacing h_{new}, i.e.,

$$\bar{p}_i^{\text{new}} = \frac{1}{h_{\text{new}}}\left[\int_{y_{i-1/2}}^{x_{j+1/2}} R(x, \bar{p})_{j-1} \, dx + \int_{x_{j+1/2}}^{y_{i+1/2}} R(x, \bar{p})_j \, dx\right], \tag{27}$$

where $h_{\text{new}} = y_{i+1/2} - y_{i-1/2}$.

Therefore the shaded areas in Fig. 14 represent the new cell average $\bar{p}^{\text{new}}$ multiplied by the new grid spacing h_{new}. In order to evaluate the integrals to a consistent order of accuracy we need to use numerical quadrature. Since the functions are polynomials the integrals can be evaluated exactly, although it does mean that the polynomials need to be interpolated at the quadrature zeros in the integral interval. This operation may therefore become quite expensive, although it is not being executed at each time step. An alternative method which is still conservative, but is only first-order accurate, is to approximate the integrals in Eq. (27) by

$$\int_{y_{i-1/2}}^{x_{j+1/2}} R(x, \bar{p})_{j-1} \, dx = \bar{p}_{j-1}^{\text{old}} \cdot (x_{j+1/2} - y_{i-1/2}).$$

This means that the new cell average is evaluated by

$$\hat{p}_i^{\text{new}} = \frac{1}{h_{\text{new}}}(\bar{p}_{j-1}^{\text{old}}(x_{j+1/2} - y_{i-1/2}) + \bar{p}_j^{\text{old}}(y_{i+1/2} - x_{j+1/2})).$$

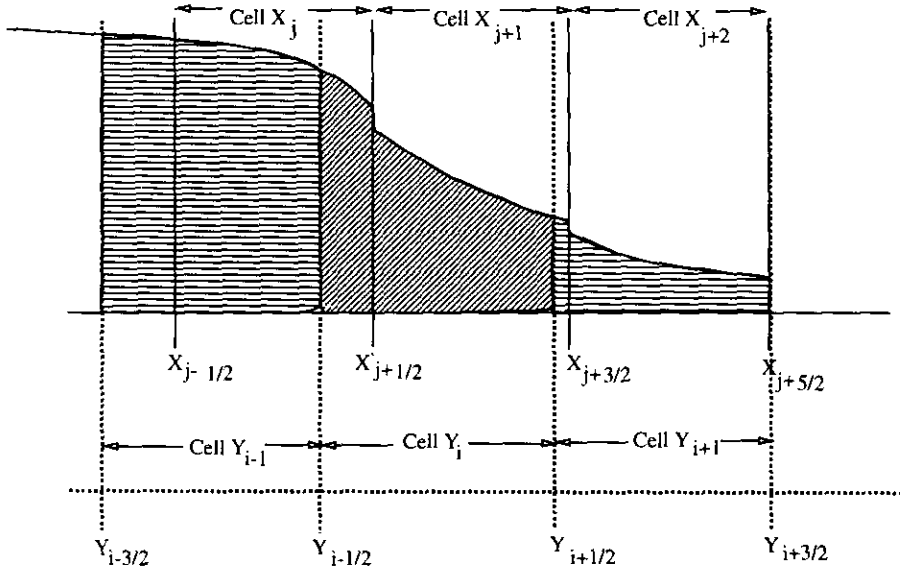


FIG. 14. Remeshing in a conservative fashion. The cells X_j are remeshed on to the cells Y_i. Shaded areas show the region of integration which must be evaluated to determine the new cell average $\bar{p}_i^{\text{new}}$.

This equation is demonstrated geometrically in Fig. 15. The new cell average is now made up of fractions of the area given by the old cell averages multiplied by the old mesh spacing. It can be appreciated that this method is easier to code since it does not involve any quadrature evaluations. Also since the operation is not evaluated at every time step the loss in accuracy may well be acceptable. The remeshing in the present calculations have been evaluated in this manner.

For a two-material configuration, where the ends of the complete domain are fixed and there is only one internal moving boundary, one mesh will be contracting whilst the other will be growing. Therefore, when it is necessary to remesh, due to CFL restrictions in the contracting domain, the growing domain can also be remeshed by adding an additional mesh point. This will help accuracy since the maximum size of h will be limited. The remeshing in the growing domain is done in a similar fashion as shown above, except the original mesh spacing will now be larger than the final mesh spacing. It should also be appreciated that in the contracting domain there may be new cells that entirely cover an original cell. In this case the operation still only involves two integrals since the integral of the completely covered cell is equal to $\bar{p}^{\text{old}} \cdot h_{\text{old}}$.

4.5. Convergence at the Moving Boundary

As has been discussed previously, for some configurations we can expect to have an expansion wave at the moving boundary connecting the physically correct boundary condition to the limit value of the slope. However, this can cause numerical problems. The nature of the ENO scheme is such that the stencil requires some ability to move in order to be essentially non-oscillatory. This clearly is not the case at the boundary where the stencil is fixed. As an example we can consider a test case where the solution has an expansion wave at the boundary. If we use the flux function given by Eq. (12), we can construct such an example by considering a configuration with slope $p''(x, t_0) = 0.3$ and an interface slope of $g' = 1.5$.

We are able to solve for a single material with a moving boundary since the physical argument that the upper material evolution cannot be dependent on the lower material evolution means that no information from the lower surface is required to solve for the upper material. Nevertheless, the interface slope $g'(x_b)$ implies the existence of a second material, although the evolution of this material is not being calculated.

Figure 16 shows the evolution of this test example at intervals of one time unit in the (x, p) plane. The left-hand figure shows the result for a fourth-order spatially accurate scheme, where the stencil order is $r = 5$. It can be seen that the expansion region develops relatively rapidly and then begins to oscillate around the correct boundary condition which is $p_b'' = -0.403846$. It should be appreciated that this test case is particularly hard to evaluate since the expansion region is moving with a similar velocity to the moving boundary. In tests where this is not the case convergence is seen in a less oscillatory fashion. The fact that the expansion region is relatively small also means that the effect on the profile in the (x, y) plane is minimal. Nevertheless, some improvement is noted if we use a lower order stencil in the last three cells next to the moving boundary. The results from using a stencil order of $r = 3$ in the last three cells can

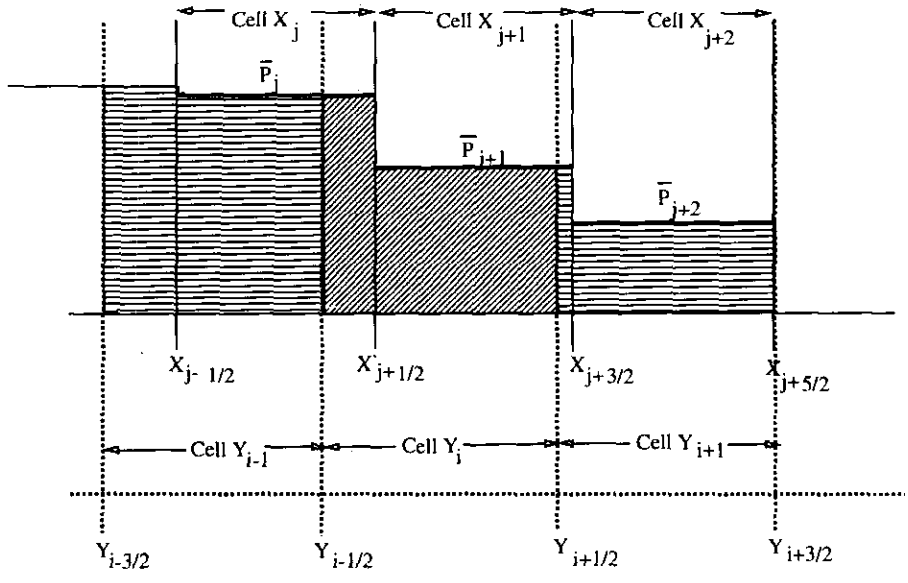
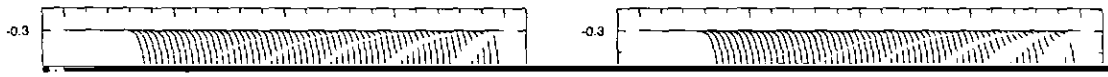


FIG. 15. Remeshing in a conservative fashion. First order accurate method. The cells X_j are remeshed on to the cells Y_i. Shaded areas show the region of integration which must be evaluated to determine the new cell average $\bar{p}_i^{\text{new}}$.


-0.3
-0.3

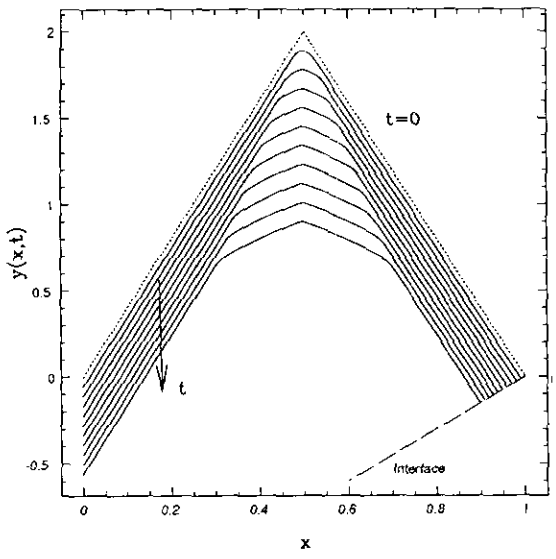


FIG. 18. Numerical solution of a Riemann problem in the (x, y) plane. Each line corresponds to the evolution after two time units. The right-hand boundary moves along an interface of slope $g'(x_b) = 1.5$.

fixed boundaries and for a domain where the boundary at $x = 1$ was moving. In the moving boundary case the boundary condition at $x = 1$ is given by the limits value $p_{\lim}$ and the boundary speed was determined from Eq. (5). In this case the minimum grid spacing was such that no remeshing took place, thereby eliminating possible errors from this operation. The interface slope was $g'(x_b) = 0$.

The solution to the problem with two fixed boundaries shows an increasing convergence rate with the accuracy of the scheme (Fig. 17). This has been seen previously in Section 3.1 since when the boundaries are fixed we only have a single material problem. Nevertheless, this acts as a good basis on which we can compare results of the moving boundary problem. Not surprisingly we see that the moving boundary problem has the same convergence rate for the first- and second-order schemes but not for the third- and fourth-order schemes. This may well be due to the use of the trapezoidal approximation to the integral in the second part of the time splitting (see Section 4.2). This approximation is, of course, only second-order accurate and is used to determine the final cell average at the new time level from which the soltion is reconstructed.

The remaining results all use the flux function $f(p)$ given by Eq. (12). The result shown in Fig. 18 is the solution to the Riemann problem solved for the fixed boundary case in Section 3.2. The initial condition is given by the uppermost line and each subsequent line corresponds to a two time unit interval. In this case we have introduced an interface with slope $g'(x_b) = 1.5$ at the right boundary. We see that the same shock information is achieved and no anomalies have been introduced due to the co-ordinate stretching.

Figure 19 shows a geometry which causes an expansion region to develop at the right-hand boundary. Each line in these figures represents the profile after an interval of 1.5 time units; the uppermost line in the (x, y) plane shows the initial condition. We note that the expansion region converges reasonably rapidly as can be seen by the right-hand side of the (x, p) plane. This expansion can be seen as a region of curvature developing in the (x, y) plane. The shock in the initial profile convects at a constant speed towards the left.

Figures 20 and 21 show the evolution of a more realistic etching configuration. Here, we see half of a symmetric

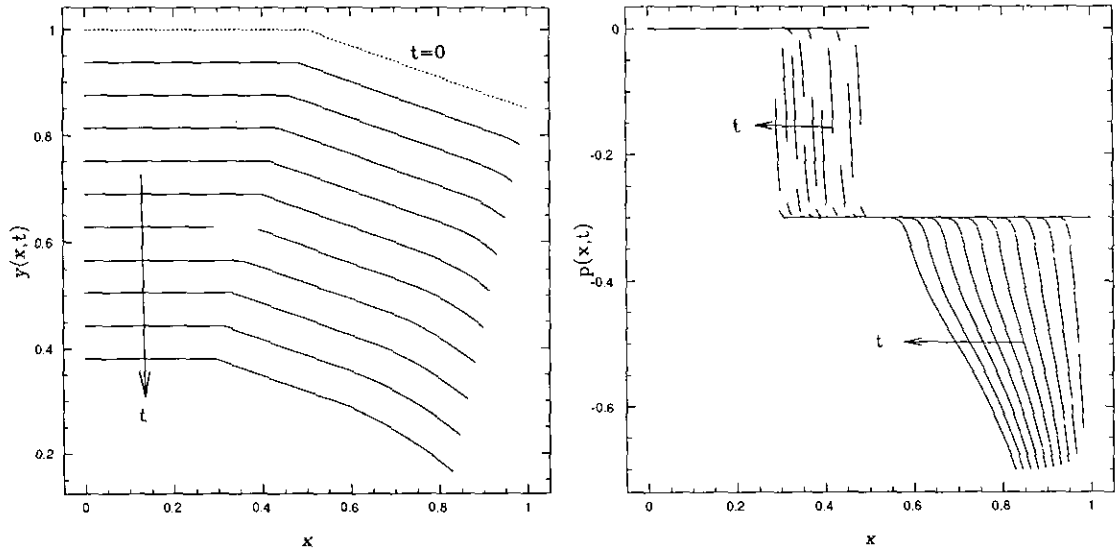


FIG. 19. Evolution of test geometry in (x, y) and (x, p) planes. The geometry is such that an expansion region develops at the right-hand moving boundary. The initial cusp in the $y(x)$ surface is seen as a shock in the (x, p) plane.

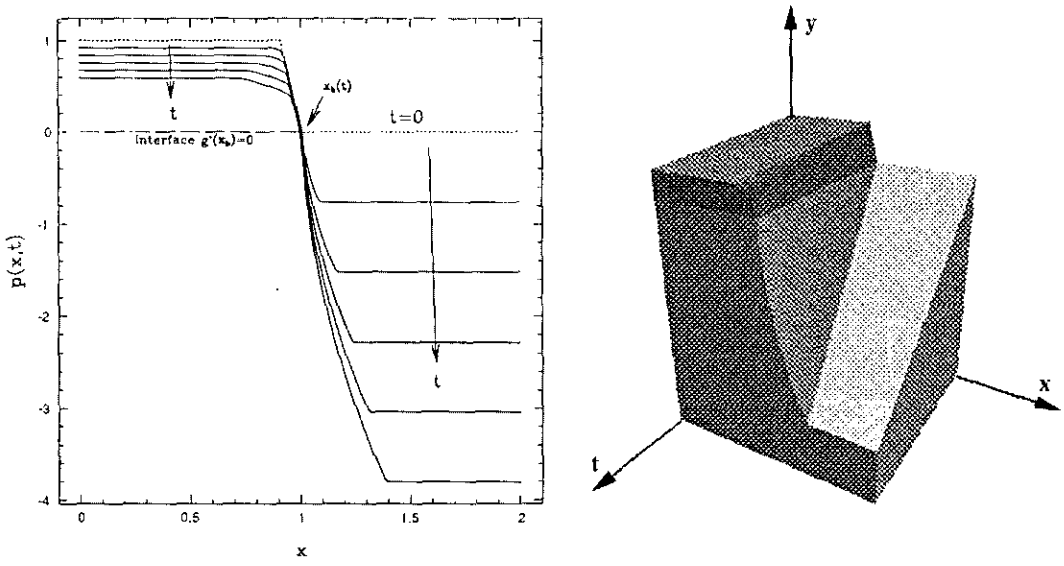


FIG. 20. Evolution of a two material configuration in the (x, y) plane with a selectivity of 10. The interface slope is $g' = 0$. We see a shock bifurcation in the upper surface and another shock development at the moving boundary in the lower surface.

configuration. The initial profile is made up of a rectangular upper surface mounted on a flat lower surface. The interface slope is therefore zero. Figure 20 shows the evolution of the surface when the selectivity is 10 whilst Fig. 21 shows the evolution when the selectivity is 20. Selectivity is the ratio of the etching rate in the lower material at zero slope ($p' = 0$) to the etching rate in the upper material at zero slope ($p^u = 0$) (i.e., selectivity $= f'(0)/f^u(0)$). Each line in the (x, y) plot represents the profile after an interval of two time units. In this case the upper material boundary condition p_b^u is the same as the limit value p_{lim}^u so there is no upper surface expansion region. Nevertheless, we see that the initial cusp in the upper material bifurcates into a left- and right-hand moving cusp which would be seen as shocks in the (x, p) plane. The lower material can be seen to develop a shock at the moving boundary. This shock connects the region of constant slope, to the right of the shock, to an expansion region. The expansion region connects the shock to the moving boundary. The calculation here assumes that the flux function in the lower material is given by Eq. (12). The effect of shadowing indicates that this is still not a very realistic form for the lower flux function (see Section 2.1). A more accurate estimate of the flux function for the lower material is addressed in the next section.

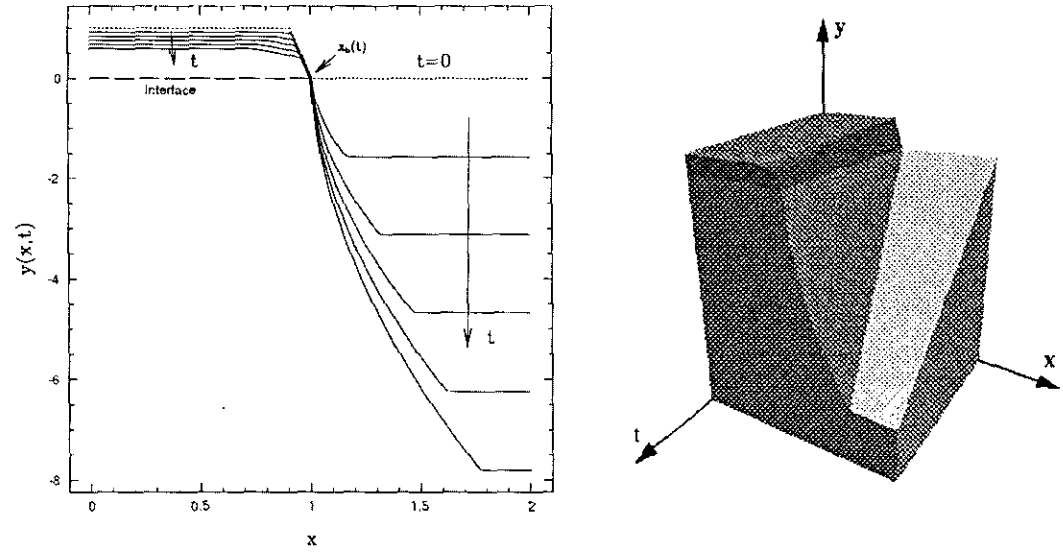


FIG. 21. Evolution of a two material configuration in the (x, y) plane with a selectivity of 20. The interface slope is $g' = 0$. We see a shock bifurcation in the upper surface and another shock development at the moving boundary in the lower surface.

6. LOWER SURFACE MODEL DEVELOPMENT

Jurgensen and Shaqfeh [1, 4] have proposed a model described in Section 2.1 which assumes that the mean-free path of the particles in the plasma is far larger than the characteristic length of the etched features. The flux function can then be calculated in the following manner.

6.1. Flux Function Calculation

We consider a flux function given by

$$f(p, B) = u_p(B) \cdot p - v_p(B),$$

where (u_p, v_p) are the local surface velocities in the (x, y) direction for a two-dimensional model. We will use (u_p, v_p) to denote the surface velocity in the co-ordinate system shown in Section 2.1 and $\mathbf{V} \equiv (u, v, w)$ to denote the surface velocity in the (x, y, z) directions for the calculation presented below. This calculation has a co-ordinate system shown in Fig. 22. The co-ordinate systems are related by a rotation of π about the x axis and so $u = u_p$ and $v = -v_p$. We note that the function B is purely dependent on the local slope p in a *locally self-shadowed* region and dependent on the position (x_{sh}, y_{sh}) in a *shadowed* region, where (x_{sh}, y_{sh}) is the position of a shadowing point. The general form of the velocity vector $\mathbf{V}$ is given by

$$\mathbf{V}(\mathbf{X}) = \int_{\Omega(\mathbf{X})} d\theta\, d\phi \sin(\theta)\, \mathbf{Y}(E, \alpha)\, \mathbf{j}(\theta, \phi). \qquad (28)$$

The nomenclature here is shown in Fig. 22. $\mathbf{Y}$ is the volumetric yield per bombarding particle which is considered to be dependent upon the energy of a particle E and the angle between its direction of travel and the surface normal at the point of impact "α" and $\mathbf{j}$ is the differential particle flux vector in the direction (θ, ϕ), where θ, ϕ define a spherical co-ordinate system. Therefore $d\theta\, d\phi \sin(\theta)$ is a differential element of solid angle. Finally, $\Omega(\mathbf{x})$ is the region of solid angle that has a line of sight of the plasma at point $\mathbf{x}$. It should be appreciated that despite the complexity of the above integral it only depends on α and the integration domain $\Omega(\mathbf{x})$, since E is assumed to be a function of (θ, ϕ). We note that α can be expressed as a function of the slope p and the particle incidence (θ, ϕ). Nevertheless, it is no trivial matter to determine the functions $\mathbf{Y}$ and $\mathbf{j}$.

Figure 23 shows a one-sided shadowed region for a point $\mathbf{X}$. The integration domain $\Omega(\mathbf{X})$ for this configuration is shown by the shaded region. We note that although this is essentially a two-dimensional configuration, since the materials are assumed infinite in the z direction, the integration domain is still three-dimensional. For this configuration, Fig. 23, where B is the angle between the shadowing point and the vertical in the (x, y) plane, the integration domain is given by

$$\Omega(\mathbf{X}) = \begin{cases} 0 \leqslant \theta \leqslant \pi/2 \\ -\pi + \phi_a \leqslant \phi \leqslant \pi - \phi_a, \end{cases}$$

where $\varphi_a = \arccos(\tan(B)/\tan(\theta))$. So it can be appreciated that, although we need to calculate the integral over a three-dimensonal solid angle, the integration domain in a two-dimensional calculation can be obtained from information in the plane of the calculation.

Shaqfeh and Jurgensen [4] have evaluated this integral, Eq. (28), for a purely energy-dependent yield. In this case we replace the volumetric yield $\mathbf{Y}(E, \alpha)$ with the average yield for a bombarding particle $\bar{Y}(\theta, \phi)$ moving in the direction defined by (θ, ϕ). This is assumed to be proportional to the average energy $\bar{E}(\theta, \phi)$ of the particle at (θ, ϕ), i.e.,

$$\bar{Y}(\theta, \phi) = k\bar{E}(\theta, \phi).$$

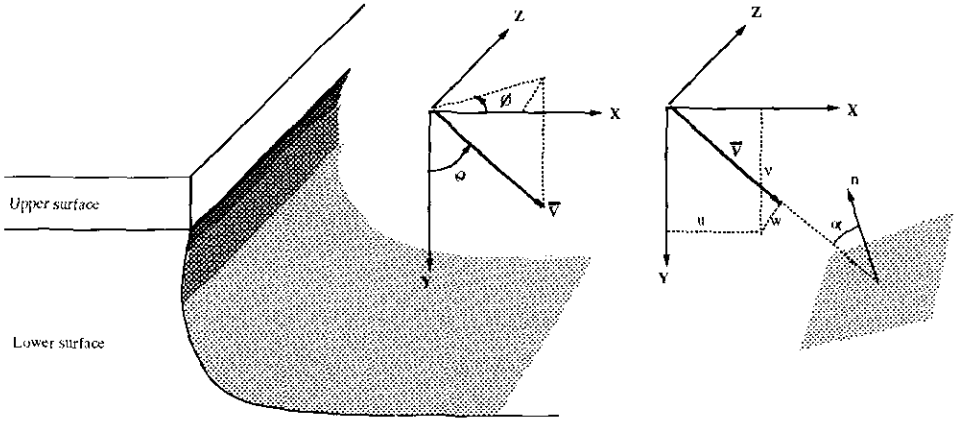


FIG. 22. Nomenclature for flux function calculation using the formulation by Jurgensen and Shaqfeh [4].

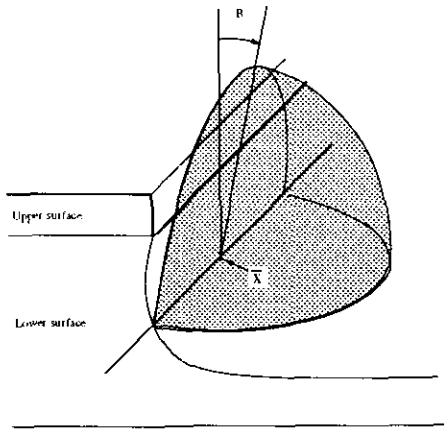


FIG. 23. Shadowing region for a point **X** in the lower surface. Here the point is only shadowed on one side by the surface interface.

We can consider the differential particle flux vector $\mathbf{j}(\theta)$ as its magnitude $j(\theta)$ times a unit vector $\mathbf{p}$ in the (θ, ϕ) direction such that

$$\mathbf{p}(\theta, \phi) = \sin(\theta)\cos(\phi)\,\mathbf{1}_x + \cos(\theta)\,\mathbf{1}_y + \sin(\theta)\sin(\phi)\,\mathbf{1}_z,$$

where $(\mathbf{1}_x, \mathbf{1}_y, \mathbf{1}_z)$ are unit lengths in the (x, y, z) directions. If we assume an azimuthal symmetry about the electric field lines which are tangential to the y direction, the velocity vector has components given by

$$u = k\int_{\Omega(\mathbf{X})} d\theta\, d\phi\, \sin^2\theta\cos\phi j(\theta)\,\bar{E}(\theta), \tag{29}$$

$$v = k\int_{\Omega(\mathbf{X})} d\theta\, d\phi\, \sin\theta\cos\theta j(\theta)\,\bar{E}(\theta). \tag{30}$$

Integrating Eqs. (29) and (30) with respect to ϕ over the integration bounds given above we obtain

$$u = -2k\int_0^{\pi/2} d\theta\, \sin^2\theta j(\theta)\,\bar{E}(\theta)\sin(\phi_a)$$

$$v = 2k\int_0^{\pi/2} d\theta\, \sin\theta\cos\theta j(\theta)\,\bar{E}(\theta)(\pi - \phi_a).$$

Here $j(\theta)$ is the magnitude of the flux of bombarding particles moving in the direction θ and $\bar{E}(\theta)$ is the average energy of those particles. The w component will necessarily be zero due to the azimuthal symmetry and the fact that there is no non-symmetric shadowing in two dimensions. To obtain distributions for these values we use the approximate distributions developed by Jurgensen [11]. His model treats the incident particles as a monoenergetic beam that is first accelerated to the average bombardment energy and then is scattered by neutral particles while crossing a field-free region with a thickness proportional to the sheath thickness. The "effective" sheath thickness is taken to be

$$D = \frac{S(1 + \frac{1}{2}\rho\sigma_{ct}S)}{2(1 + \rho\sigma_{ct}S)},$$

where S is the sheath thickness, ρ is the number density, and σ_{ct} is the charge-transfer cross section. The energy weighting factor for this model is

$$\bar{E}(\theta) = \cos^2(\theta).$$

The magnitude of the flux vector is given by

$$j(\theta) = \frac{J}{\cos(\theta)}\,[\sigma(\theta)\,\rho\,D\exp(-\rho\sigma_\theta D)],$$

where $\sigma(\theta)$ is the differential cross section and σ_θ is the total cross section for scattering at angles greater than θ. The evaluation of this term is simplified by the analytic relationship,

$$\frac{2\pi}{J}\int_0^\theta d\theta\, \sin\theta\cos\theta j(\theta) = \exp(-\rho\sigma_\theta D).$$

The right-hand side can be evaluated, given a value of ρ and D and using a small angle approximation with a modification factor for σ_θ as given in [11]. This function can then be numerically differentiated with respect to θ to approximate $[\sin\theta\cos\theta j(\theta)]$. The constants J and k can be eliminated by making the governing slope equation non-dimensional.

In practice the integrals are evaluated by tabulating them as functions of B and then using linear interpolation to determine a continuous function. To appreciate why this is possible we evaluate the velocity integral (29), (30) using a co-ordinate system shown in Fig. 24. In this case the integration domain for Fig. 23 is given by

$$\Omega(\mathbf{x}) = \begin{cases} -\pi/2 \leqslant a \leqslant \pi/2 \\ B \leqslant b \leqslant \pi/2. \end{cases}$$

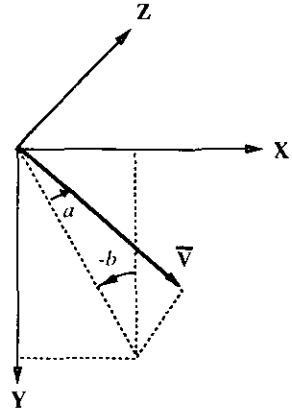


FIG. 24. Alternative co-ordinate system to evaluate the velocity integrals.

A unit vector for this system is

shadowing points exterior to the computational domain. For a configuration shown in Fig. 25, where the upper

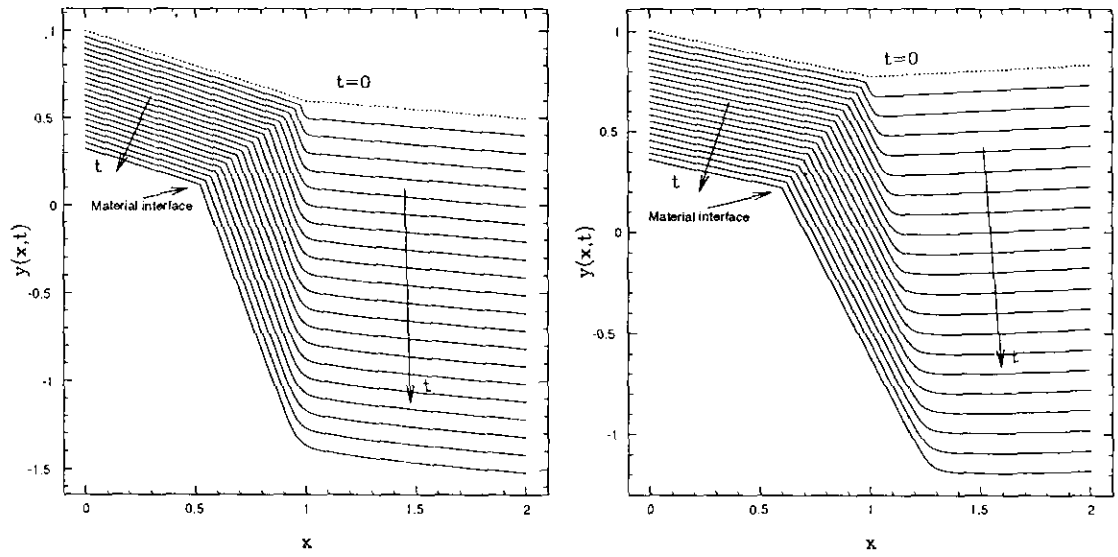


FIG. 27. Solution of a test problem, both plots show the same geometry but the second plot is solved in an axis system which is rotated by 9°. In each case the dotted lines show the initial condition and each subsequent line corresponds to an evolution of 0.05 time units. A selectivity of three was used.

shown in Fig. 26. In the original (x, y) axis system we have the slope equation,

$$p_t + f(p)_x = 0, \qquad \text{where} \quad p = \partial y/\partial x$$
$$f(p) = u_p\, p - v_p.$$

In the (r, s) axis system the slope equation is

$$q_t + f(q)_r = 0, \qquad \text{where} \quad q = \partial s/\partial r$$
$$f(q) = u_q q - v_q.$$

The slopes p and q are related by the equation

$$\arctan q = \arctan p + \alpha$$

and the local velocities are related by the equations

$$u_q = u_p \cos\alpha - v_p \sin\alpha$$
$$v_q = v_p \cos\alpha + u_p \sin\alpha.$$

Since we tabulate the values of the local surface velocities (u, v) it is quite trivial to generate the flux function for either co-ordinate system. Provided we have a region of constant slope at either end of the computational domain so that we can impose Dirichlet boundary conditions there are no further complications to calculating the surface evolution.

The first plot in Fig. 27 shows the evolution of a test case in the original axis system (x, y). Here we have a slope of $p = -0.4$ joining a slope of $p = -0.1$. The selectivity is three and the interface slope is $g'(x_b) = 1$. The lines show the evolution of the surface after 0.05 time units and the initial condition is given by the upper dotted line. The second plot shows the same configuration but is solved in an axis system which is rotated clockwise by 9°. We see the same shock development at the moving boundary in the lower material; this is perhaps more evident if we consider the solution rotated back to the original co-ordinate system as shown in Fig. 28.

6.2.3. *Outflow Buffer Domain*

The use of a rotated co-ordinate system introduces the problem that the lower material boundary is no longer a

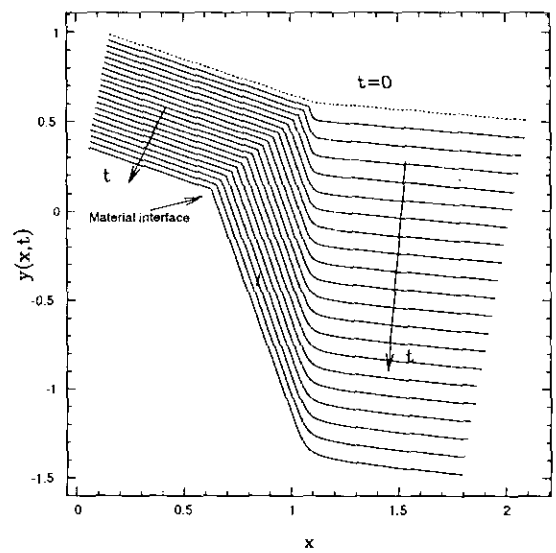


FIG. 28. Test problem solved in rotated co-ordinate system but plotted in original axis system. Initial condition is given by the dotted line and each subsequent line corresponds to an evolution of 0.05 time units. A selectivity of three was used.

point of symmetry. This is not a problem if we are only etching a single feature since we can simply extend the computational domain out to a point where the movement of the far side boundary does not interfere with the region of interest. Nevertheless, if the feature has a shadowing point from the far side we are not able to extend the computational domain in a similar fashion. To overcome this difficulty we extend the domain but use the local velocity at a physically meaningful position $\mathbf{V}(x_{bc})$ for all values beyond this point, thereby producing an outflow buffer domain, i.e.,

$$\mathbf{V}(x) = \begin{cases} \mathbf{V}(x), & x \leqslant x_{bc} \\ \mathbf{V}(x_{bc}), & x > x_{bc}. \end{cases}$$

This allows calculation up to the boundary position of the initial profile, in the unrotated co-ordinate system, provided it is an outflow boundary. The outflow condition is necessary since the extended domain is unphysical, and so information cannot travel from this region. This is not as restricting as it might seem since in general this boundary is an outflow condition; if this is not the case a solution of the complete geometry on the right-hand side would be required.

6.3. Undercutting and Reactive Ion Etching (RIE) Lag

Two physically observed phenomena that can be modeled by the algorithm presented are undercutting and reactive ion etching (RIE) lag.

Undercutting refers to the lower surface *undercutting* the upper surface. To appreciate how this can occur we first consider the flux function as calculated in Section 6.1. We note that the flux function is an increasing symmetric function of p, in the unrotated co-ordinate system, see Fig. 29. Therefore, as the slope tends to infinity so does the flux function. This happens because in calculating the local surface velocity we assume that the surface etches in the net direction of the energy flux vectors of particles hitting the surface. Therefore, even if the surface has an infinite slope it will have a finite local velocity. The only way, therefore, a surface of infinite slope can have a finite local velocity is if it has an infinite flux. From the form of the flux function it can be appreciated that there are values of the moving boundary speed which would not allow for a solution to the lower Rankine Hugoniot condition, Eq. (5). We recall that the moving boundary speed dx_b/dt is geometrically interpreted as the slope of a straight line in the $(p, f(p))$ plane representing the Rankine Hugoniot condition. In general the two materials have different etching rates as is indicated by the selectivity. Therefore, it is conceivable that the moving boundary speed, which is determined by the upper surface limit slope $p^u_{\lim}$, may not have a root in the lower surface Rankine Hugoniot condition. Nevertheless, this does not mean that there is not a physically meaningful solution, but that the equation in this co-ordinate system does not have a solution. If we now consider the flux function for a co-ordinate system which is rotated by 9°, as shown by the dashed line in Fig. 29, we see that there will be solutions for a wider range of moving boundary speeds since a given line can have a larger range of slope values that intersect this curve. The kink in this flux curve at approximately $p = 6$ corresponds to the infinite slope ($p = \infty$) in the original unrotated co-ordinate system since $a\tan(6.314) + 9 \approx 90$. At this point, the lower surface undercuts the upper surface. It can be appreciated that calculation of this boundary would have been impossible without the use of a co-ordinate rotation in the computational domain. We find

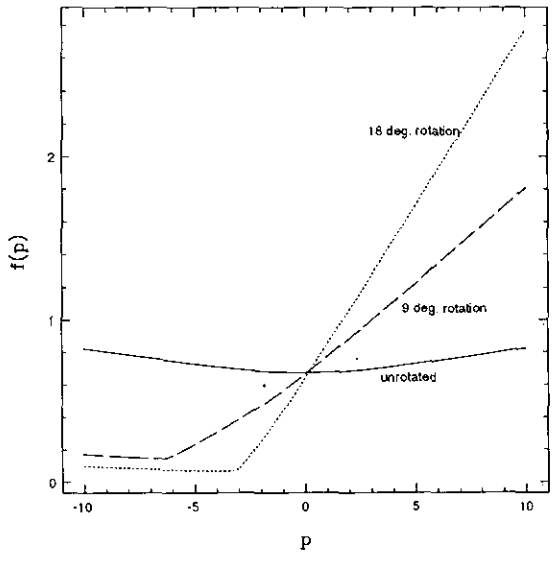


FIG. 29. Angle independent flux functions for solution in rotated domains. For an angle independent flux function we see that in the unrotated co-ordinate system the flux function tends to infinity as $|p| \to \infty$.

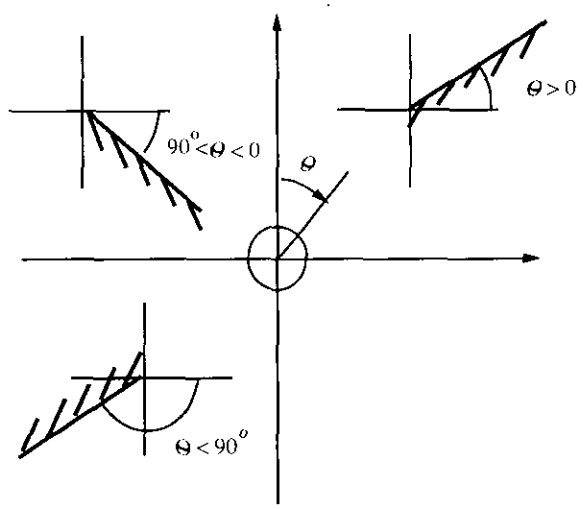


FIG. 30. Interpretation of the $\theta = \arctan(p)$ angle used in flux function calculation.

that if we use the same form for the flux function in both the upper and lower surface then for an upper limit slope value of $p_{\lim} = 10$ we can solve up to a selectivity of about six. However, if we use a rotation of 18°, as shown by the dotted line in Fig. 29, then a selectivity of the order of 10 can be achieved.

It is worth noting at this point that we require information about the local surface velocity in three quadrants in order to use the present algorithm. As can be seen in Fig. 30 we tabulate values for surface angles between $-180° \leqslant \theta \leqslant 90°$. This is because we require the region $-180° \leqslant \theta \leqslant -90°$ to determine the lower surface boundary condition in the case of undercutting. In this case the computational angle in the rotated domain will be close to $-90°$ and when the physically unrotated angle is recovered it may be less than $-90°$. We calculate the positive quadrant $0 \leqslant \theta \leqslant 90°$ for stability considerations since we wish to consider the surface where $p = 0$ and therefore slight positive slope values are plausible in the calculation. In the case where we have two shadowing points, the far side shadowing point would usually have a positive θ value. In this case we set $\theta = -180° + \theta$ since this is the equivalent shadowed angle for the flux calculation.

The second phenomenon that we wish to capture is the RIE lag. It has been observed that features of different aspect ratios etch at different rates. Aspect ratio is defined as the depth of the planarising layer (the lower surface) divided by the width of a trench. In general, we find that features in an open field, where the aspect ratio is zero, etch at a faster rate than features in close proximity. This is to be expected as when features are in close proximity there are two shadowing points. Thus the integration domain used to calculate the local surface velocity (and therefore the flux function) will be smaller.

Results for both these phenomena are shown in Section 6.4.2.

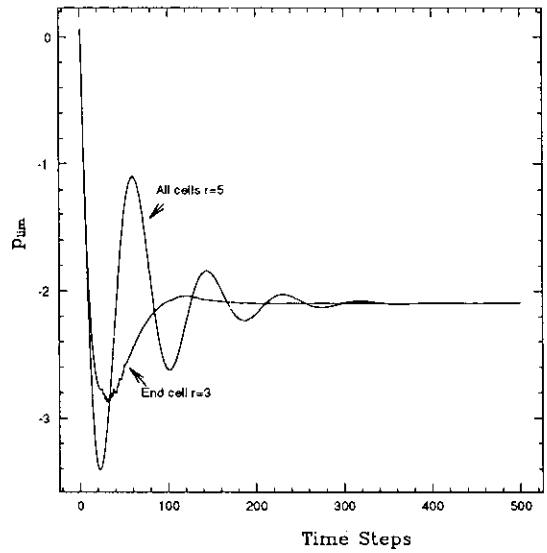


FIG. 31. Lower surface boundary slope $p_{\lim}$ convergence at the moving interface for case shown in Fig. 27. Plot shows convergence using a five-point stencil with and without lower order stencils at the boundary.

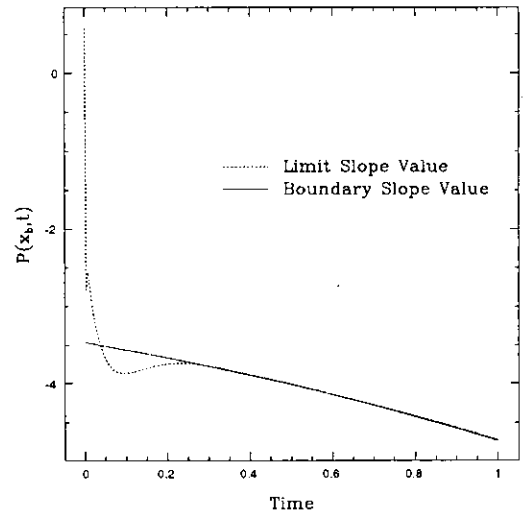


FIG. 32. Lower surface boundary convergence at the moving interface for the case shown in Fig. 36. The solid line shows the numerical slope boundary condition p_b^l and the dotted line shows the limit slope value $p_{\lim}^l$.

6.4. Results

6.4.1. *Lower Surface Boundary Condition Convergence*

As previously mentioned the lower surface boundary condition at the moving interface can also cause complications. In a similar fashion to the upper surface boundary condition, Section 4.5, we use a lower order stencil at the moving

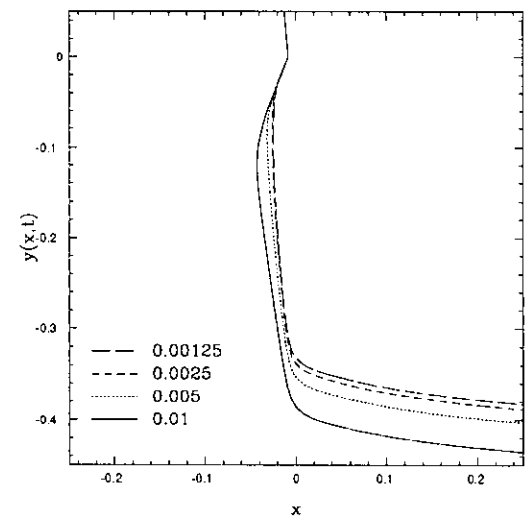


FIG. 33. Etched profiles after a non-dimensional time of 0.4 for the case shown in Fig. 34. As the spatial resolution is increased in the lower surface we see convergence to a unique profile.

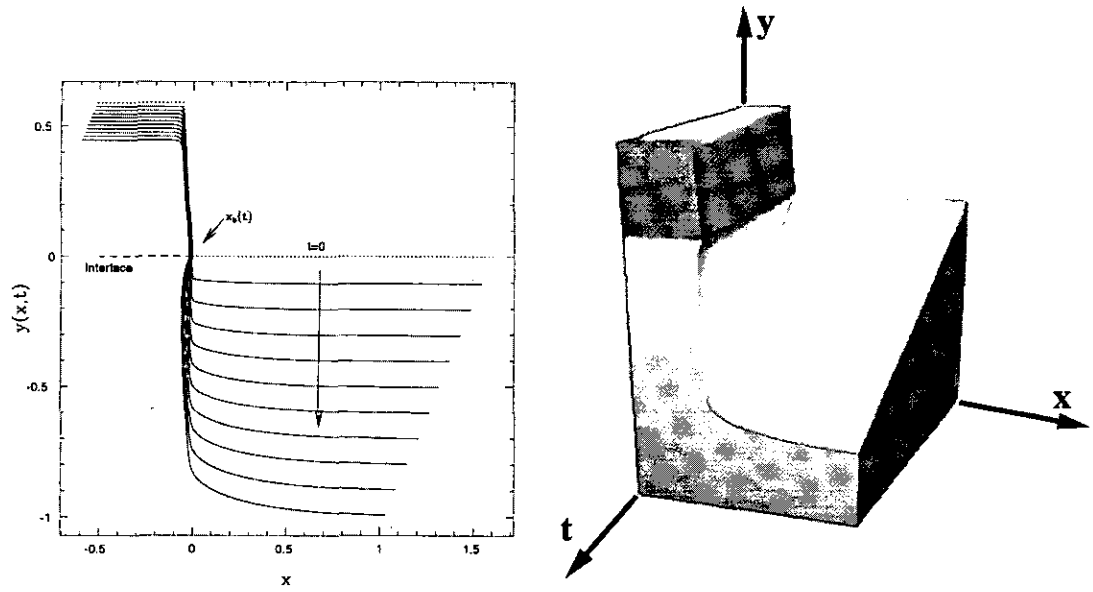


FIG. 34. Evolution of two material configurations in the (x, y) plane with a selectivity of seven; each line corresponds to a non-dimensional time of 0.1. The lower surface is only shadowed from the left. We see that there is strong undercutting of the upper surface.

boundary. This is illustrated in Fig. 31. Here we see the convergence of the lower surface limit value p^l_{lim} to the appropriate boundary condition p^l_b for the test case shown in Fig. 28. We see results for two cases; the first corresponds to using a five-point stencil in all the computational domain of the lower surface and the second uses a three-point stencil in the three cells adjacent to the moving boundary. As can be seen the convergence is far more oscillatory if we use a five-point stencil everywhere. The effect is damped by the introduction of lower order stencils at the boundary. This is probably due to the addition of more numerical viscosity in this region.

Figure 32 shows the boundary convergence of the test case shown in Fig. 36. Due to the far side shadowing the boundary value has a time dependence. The dotted line shows the limit slope value, p^l_{lim}, with time whilst the solid

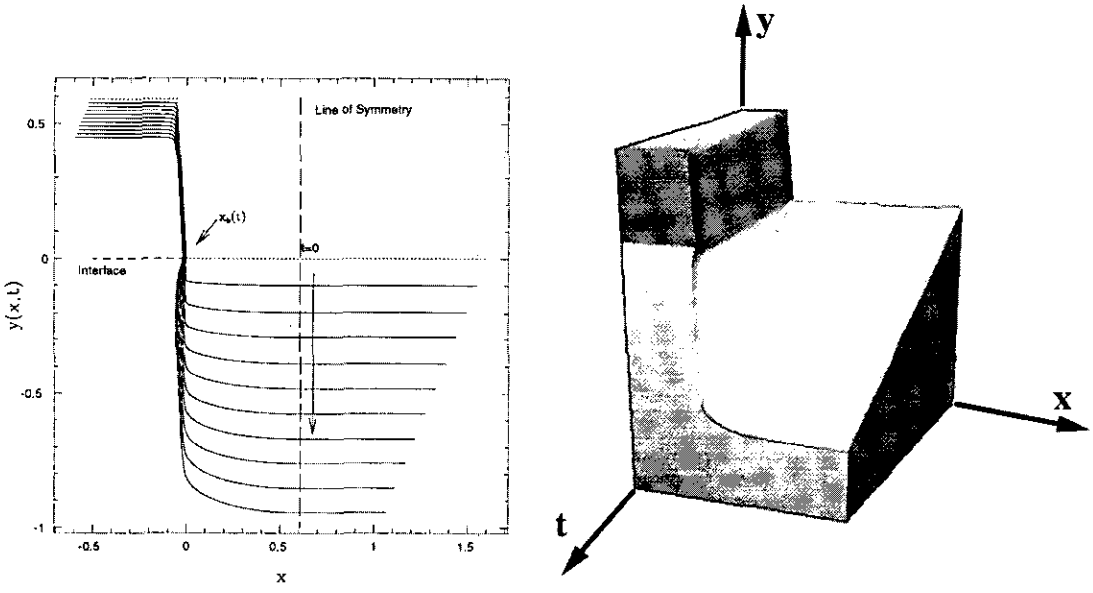


FIG. 35. Evolution of a two-material configuration in the (x, y) plane with a selectivity of seven; each line corresponds to a non-dimensional time of 0.1. The physical domain is assumed to be symmetric about the marked line which means that there is shadowing from both sides in the lower surface.

line shows the boundary slope value with time, p_b'. As can be seen there is an initial region where the limit value

bias voltage and rf frequency are 500 V and 13.5 MHz, respectively. The upper surface is assumed to be locally self-

7. CONCLUDING REMARKS

We have developed a numerical algorithm to solve a nonlinear hyperbolic conservation law on a moving mesh. Two approaches to formulating the flux function have been adopted: the first is used in the upper material where the surface is considered as locally self-shadowed and the second is used in the lower material where the material can also be self-shadowed. In either case the major complications result from the moving interface between the materials. This is because expansion waves and shocks may develop at this point. This has caused complications because the ENO reconstruction algorithm requires mesh points on either side of the shock or expansion region in order to resolve it satisfactorily. This is not possible at the moving boundary and we therefore find convergence problems which result in a slow or oscillatory convergence. We note that the algorithm presented for the two-material configuration appears to be only second-order accurate. Therefore the use of a high-order accurate scheme such as ENO may not be suitable and the use of a second-order scheme which does not require a moving stencil could be more appropriate.

To extend the lower surface flux model we ideally need to introduce the angle dependence of the yield function. We also require better knowledge of the energy and angle distribution of the incident particles. Knowledge of the particle energy and angle distribution is beyond the scope of this work and at present we consider this as a model input. The volumetric yield for a given particle energy and incidence is also assumed and is required as a model input. However, if we assume knowledge of these inputs we can tabulate the local surface velocities as a function of the shadowing angle B for a given slope angle p. We recall that the slope allows us to determine the angle α which is the only extra information required to evaluate the velocity integral, Eq. (28), assuming the model inputs stated above. Nevertheless, this requires $O(N^2)$ storage, where N is the number of tabulated points for the B shadowing angle. This of course assumes that we wish to use a comparable resolution in the tabulated slope values as the shadowing angle, which may not always be the case. However, the storage requirements are noticeably increased.

Finally, it should be appreciated that the algorithm can be solved in parallel since, once the moving boundary has been propagated forward over a time step, the interior domains can be solved independently. There is a coupling between the surfaces due to the shadowing angle calculation, although at present this calculation is only made at the beginning of each time step. If this calculation was updated during the time step the algorithm could still be solved in parallel since the solution of the upper surface is independent of the lower surface, if it is locally self-shadowed. Therefore, this allows the upper surface to be propagated a time step ahead of the lower surface which provides sufficient information to solve the lower surface shadowing angle in the previous time step.

ACKNOWLEDGMENTS

We thank Dr. Charles Jurgensen, from AT&T Bell laboratories, for his insight into the physical problem and the experimental information that he has provided. Our gratitude goes to Professor Eytan Barouch for proposing this project and providing the founding papers. This work was supported by DARPA under U.S. Army Contract DAAL01-92-K-0248 and by the N.S.F. under Grant ECS-90-23760.

REFERENCES

1. C. W. Jurgensen and E. S. G. Shaqfeh, *J. Vac. Sci. Technol. B* **7**, 1488 (1989).
2. J. C. Arnold and H. H. Sawin, *J. Appl. Phys.* **70**, 5314 (1991).
3. D. S. Ross, *J. Electrochem. Soc.* **135**, 1235 (1988).
4. E. S. G. Shaqfeh and C. W. Jurgensen, *J. Appl. Phys.* **66**, 4664 (1989).
5. J. Pelka, K. P. Muller, and H. Mader, *IEEE Trans. Comput. Aided Des.* **7**, 154 (1988).
6. R. E. Lee, *J. Vac. Sci. Technol. B* **16**, 164 (1979).
7. D. S. Ross, *J. Electrochem. Soc.* **135**, 1260 (1988).
8. A. Harten, B. Engquist, S. Osher, and S. R. Chakravathy, *J. Comput. Phys.* **71**, 231 (1987).
9. C. Shu and S. Osher, *J. Comput. Phys.* **83**, 32 (1989).
10. E. M. Rønquist and A. T. Patera, *Int. J. Numer. Methods Eng.* **24**, 2273 (1987).
11. C. W. Jurgensen, *J. Appl. Phys.* **64**, 590 (1988).
12. S. J. Sherwin, E. Barouch, G. E. Karniadakis, and S. A. Orszag, *J. Vac. Sci. Technol. B* **11**, 1310 (1993).