GAKUTO International Series

Mathematical
Sciences
and
Applications

Volume 1

Proceedings of International Conference:

Nonlinear Mathematical Problems in Industry I

H. Kawarada
N. Kenmochi
N. Yanagihara

September 1993

Gakkōtosho, Tokyo, Japan

GAKUTO International Series, Mathematical Sciences and Applications, Vol. 1(1993)
Nonlinear Mathematical Problems in Industry, pp.281-292

Parallel Computing Used in Solving Wet Chemical Etching Semiconductor Fabrication Problems

J. C. Bruch, Jr., C. A. Papadopoulos and J. M. Sloss

Abstract

Wet chemical etching is an important process in the manufacturing of microelectronic devices. This technique removes material selectively from the surface of a solid body by the application of caustic fluids. Predicting and understanding the etching profile growth (a moving boundary) is therefore very useful. A mathematical model describing this process can be formulated in terms of a partial differential equation valid in a time dependent, *a priori* unknown domain. Depending upon the magnitude of the physical parameters of the problem, a variational inequality formulation can be obtained valid in a fixed domain. An iterative numerical scheme with projection has been developed for this latter problem and implemented on an iPSC/860 parallel computer. The highly efficient parallel algorithm, some numerical results and other computational details are discussed for a representative example.

1 Introduction

Wet chemical etching ([2] - [5], [7]) is an important technique in the fabrication of semiconductors. In this process, an impermeable mask is placed on some part of the surface of the body to be etched. Caustic fluids are then allowed to come into contact with the body and etch into the body wherever the mask is not present. As time proceeds, more and more of the body is cut away by the fluids. Thus, this problem falls into the class of moving boundary problems. A mathematical model describing this process can be formulated in terms of a partial differential equation valid in a time dependent,*a priori* unknown domain. During this process, undercutting, or removal of solid material from under the mask, occurs. If the area of undercutting is too large, it can cause short-circuits and other problems with the integrity of the semiconductor. Thus it is important to describe this behavior as well as possible. Predicting the profile of the moving boundary over time would allow improved manufacturing of semiconductors.

Vuik and Cuvelier [8] presented a representative example of an etching problem of this type. Their numerical results were obtained using the finite element method on a serial computer. They considered two cases of the problem. One for which the Sherwood number is assumed to be large and the other where it is not. The latter required a moving mesh approach. The first case is the one pursued herein. With these assumptions a variational inequality formulation can be found in a fixed domain. This fixed domain problem will be solved using a finite difference scheme with projection. See Crank [1] for references to researchers who have solved problems of this type using finite differences with projection for the serial case. Wang [9] has developed a parallel SOR algorithm with projection and has applied it to a free boundary seepage problem. Herein this algorithm will be adapted to the moving boundary problem. This problem has a more interesting geometry as well as the fact that the derived partial differential equation (PDE) has a step discontinuity in the solution region. Details of how these difficulties were handled are described in Section 4. It was found that the problem parallelized extremely well, and the results of Vuik and Cuvelier were reproduced with reasonable accuracy although different methods of solution were used.

The outline of the paper is as follows: In Section 2 we shall introduce the physical problem. In Section 3 the mathematical model and derivation of the complementarity system is described. In Sections 4.1 and 4.2 the numerical algorithm and the parallel scheme are described, respectively, and in Section 4.3 numerical results are presented. From the numerical results we see that the method parallelizes well and that acceptable speedups are maintained from 1 to 64 nodes on the iPSC/860 architecture. With the large amount of computing time required by this problem, it becomes clear that efficient parallel numerical algorithms for the new architectures, such as this, must be developed.

2 Physical Problem

Chemical etching has been used for many different types of applications, for example the manufacture of television tubes, production of lasers, production of semiconductors and any other process where fine depth control of a physical body is to be performed. Here we shall look at the case of semiconductor fabrication. This is the physical problem that was considered by Vuik and Cuvelier [8]. Following their physical model, we shall place two photoresist layers (masks) onto a flat plate of length L (Figure 1). We shall allow a gap of length $2a$ between the two layers and caustic fluid to flow on top of the plate. Further, we shall assume that only one component of the caustic fluid determines the process, there is no convection in the etching medium, and that the etching process is isotropic. We shall also assume that the thickness of the mask is infinitely thin and that L is large in comparison with a.

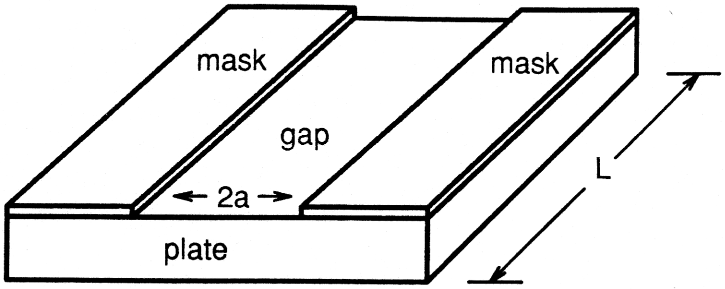

Figure 1: Physical problem.

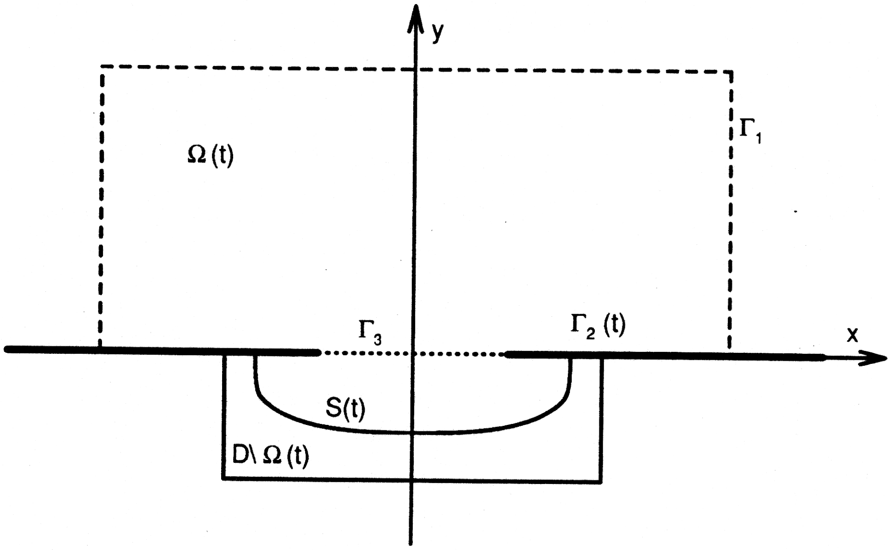

Figure 2: Side view of physical problem showing mathematical solution setup.

The latter allows us to treat the problem as two dimensional, and we shall consider the cross section (Figure 2) of our original problem. We shall let $\Omega(0)$ be a square region in the caustic fluid which is large enough so that increasing the size of $\Omega(0)$ will not change the etching process. We shall let D_1 be a rectangular region in the plate's cross section such that all the etching will occur in D_1 for $t \in [0, T]$. Denote by Γ_3 the slit of length $2a$, $\Gamma_3 = \partial\Omega(0) \cap \partial D_1$. Then the domain of the problem, D, is the union of the upper rectangular region $\Omega(0)$, the lower rectangular region D_1 in the plate, and Γ_3 along the boundary between the two, i.e. $D = \Omega(0) \cup D_1 \cup (\partial\Omega(0) \cap \partial D_1)$. Let $\Omega(t)$ be the open region in D at time $t \in (0, T)$ that is wet. Denote by Γ_1 the boundary of $\Omega(0)$ that is not along the mask or Γ_3, and $\Gamma_2(t)$ for the upper and lower surfaces of the mask that are exposed to the caustic fluid. This is the problem that we shall solve numerically using a parallel supercomputer.

3 Mathematical Model

3.1 Dimensionless Model

We shall start with the dimensionless form of the above problem when the Sherwood number is assumed to be large. Following Vuik and Cuvelier [8] the problem becomes: Find the concentration function $C = C(x,y,t)$ in $\Omega(t)$ and the moving boundary $S(t)$ such that

$$(1) \qquad \frac{\partial C}{\partial t} - \Delta C = 0 \text{ in } Q_\Omega = \{(x,y,t) \mid (x,y) \in \Omega(t), t \in (0,T)\}$$

with initial condition

$$(2) \qquad C(x,y,0) = 1 \text{ in } \Omega$$

and boundary conditions

$$\begin{array}{lrcl}
(3) & C & = & 1 \text{ on } \Gamma_{1t} = \Gamma_1 \times (0,T) \\
(4) & \frac{\partial C}{\partial \bar{n}} & = & 0 \text{ on } \Gamma_{2t} = \{(x,y,t) \mid (x,y) \in \Gamma_2(t), t \in (0,T)\} \\
(5) & C & = & 0 \text{ on } S_t(t) = \{(x,y,t) \mid (x,y) \in S(t), t \in (0,T)\} \\
(6) & \frac{\partial C}{\partial \bar{n}} & = & -Bv_n \text{ on } S_t(t)
\end{array}$$

where $\bar{n}$ denotes the unit normal vector on $\Gamma_2(t)$ pointing outward with respect to $\Omega(t)$, v_n is the normal velocity of the boundary $S(t)$, and $B = \frac{\hat{D}}{\sigma C_0}$ ($\hat{D}$ is the diffusion coefficient, σ is a material constant, and C_0 is the initial concentration). Although one can also formulate the problem when the magnitude of the Sherwood number is small as well (Vuik and Cuvelier [8]), it will not lead to a fixed domain formulation similar to the one that follows.

3.2 Fixed Domain Formulation

The following fixed domain method using a Baiocchi type transformation will transform the problem from a moving boundary problem to one with fixed boundaries. It is this problem that will be solved numerically in the next section. The following formulation of this problem is equivalent to the one-phase Stefan problem which has an equivalent variational formulation [6]. For the fixed domain formulation we shall replace (2) - (4) by:

$$\begin{array}{lrcl}
(7) & C & = & C_0(x,y) \text{ in } \Omega(0) \\
(8) & C & = & g(x,y,t) \text{ on } \Gamma_t = \Gamma_{1t} \cup \Gamma_{2t}
\end{array}$$

where we assume from the physical conditions that the functions C and g are nonnegative, and that B is a strictly positive constant.

Define an extension $\tilde{C}$ of C to a domain $Q_D = D \times (0,T)$, where D is the domain that is large enough to strictly contain $\Omega(t)$ for $t \in (0,T)$, by:

$$\tilde{C}(x,y,t) = \begin{cases} C(x,y,t) & \text{in } Q_\Omega \\ 0 & \text{in } Q_D \setminus Q_\Omega \end{cases} \tag{9}$$

and in a similar manner extend g and $C_0(x,y)$, more precisely:

$$\tilde{g}(x,y,t) = \begin{cases} g(x,y,t) & \text{on } \Gamma_t \\ 0 & \text{on } (\partial D \times (0,T)) \setminus \Gamma_t \end{cases} \tag{10}$$

$$\tilde{C}_0(x,y) = \begin{cases} C_0(x,y) & \text{in } \Omega(0) \\ 0 & \text{in } D \setminus \Omega(0). \end{cases} \tag{11}$$

With these extensions, $\tilde{C}$ satisfies in a distributional sense ([6]):

$$\frac{\partial \tilde{C}}{\partial t} - \Delta \tilde{C} = -B\frac{\partial \chi_\Omega}{\partial t} \text{ in } Q_D \tag{12}$$

$$\tilde{C} = \tilde{g} \text{ on } \partial D \times (0,T) \tag{13}$$

$$\tilde{C}(x,y,0) = \tilde{C}_0(x,y) \text{ in } D \tag{14}$$

where χ_Ω is defined to be the characteristic function for Q_Ω in Q_D. Then as in Vuik and Cuvelier [8] the following Baiocchi transformation is introduced:

$$w(x,y,t) = \int_0^t \tilde{C}(x,y,\tau)d\tau \quad (x,y) \in D, \quad t \in (0,T) \tag{15}$$

and if we integrate equation (12) with respect to t, then

$$\frac{\partial w}{\partial t} - \Delta w = \tilde{C}_0(x,y) + B(\chi_\Omega(x,y,0) - \chi_\Omega(x,y,t)). \tag{16}$$

This leads to the following complementarity system :

$$\frac{\partial w}{\partial t} - \Delta w - f \geq 0 \text{ in } Q_D \tag{17}$$

$$w \geq 0 \text{ in } Q_D \tag{18}$$

$$\left(\frac{\partial w}{\partial t} - \Delta w - f\right) w = 0 \text{ in } Q_D \tag{19}$$

$$w(x,y,t) = \int_0^t \tilde{g}(x,y,\tau)d\tau = G(t) \quad \text{on } \partial D \times (0,T) \tag{20}$$

$$w(x,y,0) = 0 \text{ in } D \tag{21}$$

where $f(x,y) = \tilde{C}(x,y,0) - B(1 - \chi_\Omega(x,y,0))$ is independent of time. If w is a solution of the above problem, then $\tilde{C} = \frac{\partial w}{\partial t}$ solves the original problem. Associated with this complementarity system is a variational inequality which Vuik and Cuvelier formulated and solved numerically for the case of a large Sherwood number. They went on and showed that there exists an unique and strong solution of the etching problem such that the moving boundary separates the regions: $\{(x,y,t) \in Q_D \mid w(x,y,t) > 0\}$ and $\{(x,y,t) \in Q_D \mid w(x,y,t) = 0\}$ if $\tilde{C_0} \in L_2(D)$ and $G \in L_2(0,T;H^{1/2}(\partial D))$. They continue on with the finite element solution of the variational inequality. Herein a parallel adaptation of Wang's scheme [9] for this problem is considered.

4 Numerical Scheme and Results

4.1 Numerical Algorithm

An adaptation of Wang's [9] iterative parallel SOR method will now be applied to (17) - (19). The regions $\Omega(0)$ and D_1 will be discretized and the problem solved using a finite difference SOR scheme with projection. For the non-parallelized case the scheme will solve the problem pointwise from right to left and top to bottom with an exception at the interface between $\Omega(0)$ and D_1 which will be discussed below. We shall denote by $(w_1)_{i,j,k}$ the i,j position of the solution in region $\overline{\Omega(0)}$ at time step k where i and j are the column and row numbers, respectively. Similarly $(w_2)_{i,j,k}$ denotes the i,j position of the solution in the region $\overline{D_1}$ at time step k. We shall let maxrow1 denote the maximum number of rows, and maxcol1 the maximum number of columns in $\overline{\Omega(0)}$. Similarly for maxrow2 and maxcol2 in $\overline{D_1}$. The reason that the region must be divided into two regions is the jump discontinuity in the differential equation (16). To concentrate our computation in the region of interest, namely $\overline{D_1}$, the bottom mesh will be 1/4 the size of the top mesh. The basic numerical algorithm is:

$$\begin{aligned}(w_1)^{(n+1/2)}_{i,j,k} = & \left(\frac{1}{\Delta t}+\frac{2}{(\Delta x)^2}+\frac{2}{(\Delta y)^2}\right)^{-1}\Bigg\{C_0+\frac{1}{\Delta t}(w_1)_{i,j,k-1}\\ & + \frac{1}{(\Delta x)^2}\left[(w_1)^{(n+1)}_{i-1,j,k}+(w_1)^{(n)}_{i+1,j,k}\right]\\ & + \frac{1}{(\Delta y)^2}\left[(w_1)^{(n+1)}_{i,j-1,k}+(w_1)^{(n)}_{i,j+1,k}\right]\Bigg\}\end{aligned} \tag{22}$$

with

$$(w_1)^{(n+1)}_{i,j,k} = (w_1)^{(n)}_{i,j,k} + \theta[(w_1)^{(n+1/2)}_{i,j,k} - (w_1)^{(n)}_{i,j,k}] \tag{23}$$

in $\Omega(0)$ and

$$\begin{aligned}(w_2)^{(n+1/2)}_{i,j,k} = & \left(\frac{1}{\Delta t}+\frac{32}{(\Delta x)^2}+\frac{32}{(\Delta y)^2}\right)^{-1}\Bigg\{-B+\frac{1}{\Delta t}(w_2)_{i,j,k-1}\\ & + \frac{16}{(\Delta x)^2}\left[(w_2)^{(n+1)}_{i-1,j,k}+(w_2)^{(n)}_{i+1,j,k}\right]\\ & + \frac{16}{(\Delta y)^2}\left[(w_2)^{(n+1)}_{i,j-1,k}+(w_2)^{(n)}_{i,j+1,k}\right]\Bigg\}\end{aligned} \tag{24}$$

with

$$(w_2)^{(n+1)}_{i,j,k} = \max\{0, (w_2)^{(n)}_{i,j,k} + \theta[(w_2)^{(n+1/2)}_{i,j,k} - (w_2)^{(n)}_{i,j,k}]\} \tag{25}$$

in D_1, where use was made of a first order backwards difference formula for $\frac{\partial w}{\partial t}$, second order central difference formulas for Δw, n denotes the SOR iterate, and θ is the SOR relaxation factor. Since there is a line of symmetry along the y-axis through the middle of the region, only the right half of the region will be solved numerically. For the points along the line of symmetry we shall use reflection and a second order central difference to obtain the additional equation:

$$\begin{aligned}(w_1)^{(n+1/2)}_{i,j,k} = & \left(\frac{1}{\Delta t} + \frac{2}{(\Delta x)^2} + \frac{2}{(\Delta y)^2}\right)^{-1} \Bigg\{ C_0 + \frac{1}{\Delta t}(w_1)_{i,j,k-1} \\ & + \frac{2}{(\Delta x)^2}(w_1)^{(n)}_{i-1,j,k} \\ & + \frac{1}{(\Delta y)^2}\left[(w_1)^{(n+1)}_{i,j-1,k} + (w_1)^{(n)}_{i,j+1,k}\right]\Bigg\}\end{aligned} \tag{26}$$

for $\Omega(0)$ and the corresponding equation for D_1 with w_2, $\frac{\Delta x}{4}$, $\frac{\Delta y}{4}$ and $-B$ replacing w_1, Δx, Δy and C_0, respectively. Also along the top and bottom of the mask, $\Gamma_2(t)$, we use a second order central difference for the normal derivative, i.e.

$$\begin{aligned}(w_1)^{(n+1/2)}_{i,j,k} = & \left(\frac{1}{\Delta t} + \frac{2}{(\Delta x)^2} + \frac{2}{(\Delta y)^2}\right)^{-1} \Bigg\{ C_0 + \frac{1}{\Delta t}(w_1)_{i,j,k-1} \\ & + \frac{1}{(\Delta x)^2}\left[(w_1)^{(n+1)}_{i-1,j,k} + (w_1)^{(n)}_{i+1,j,k}\right] \\ & + \frac{2}{(\Delta y)^2}(w_1)^{(n+1)}_{i,j-1,k}\Bigg\}\end{aligned} \tag{27}$$

for $\Omega(0)$ and the corresponding equation for D_1 with w_2, $\frac{\Delta x}{4}$, $\frac{\Delta y}{4}$ and $-B$ replacing w_1, Δx, Δy and C_0, respectively along with $(w_2)^{(n+1)}_{i,j+1,k}$ replacing $(w_1)^{(n+1)}_{i,j-1,k}$ in the last term. Equations (23) and (25) are also used in conjunction with equations (26) and (27), respectively. For the remainder of the boundary conditions we set:

$$(w_1)^{(n+1)}_{1,j,k} = t \tag{28}$$

$$(w_1)^{(n+1)}_{i,1,k} = t \tag{29}$$

$$(w_2)^{(n+1)}_{1,j,k} = 0 \tag{30}$$

$$(w_2)^{(n+1)}_{i,maxrow2,k} = 0 \tag{31}$$

At the interface we first solve for $(w_1)^{(n+1)}_{i,maxrow1,k}$ along the top of the mask using (27) until we reach $a_1 - 1$, where a_1 is the integer number of the mesh point where the mask terminates and the gap begins. Starting at a_1, solve for $(w_1)^{(n+1)}_{i,maxrow1,k}$ on Γ_3 using an average of the values of w above and below, namely:

$$(w_1)^{(n+1/2)}_{i,maxrow1,k} = \frac{1}{2}\left[(w_1)^{(n)}_{i,maxrow1-1,k} + (w_2)^{(n)}_{i,5,k}\right] \tag{32}$$

Then we start at the left and work towards the right for the first row of $\overline{D_1}$. In the etched region up to the end of the mask, point numbered a_2, we use a linear interpolation of the values of (32) on Γ_3 to fill in the smaller mesh. Then starting at $a_2 - 1$ we use equation (27) to find $(w_2)^{(n+1/2)}_{i,1,k}$ along the underside of the mask. For rows 2 through maxrow2-1 we proceed as usual from right to left. This is the algorithm that was used to solve the problem serially.

4.2 Parallel Scheme

To adapt Wang's [9] parallel scheme to the above algorithm it will be necessary to divide up the region into horizontal strips, subregions, (see Figure (3)). The fact that there is a jump in the second derivative along Γ_3 in the slit made it necessary to put a division there. Each subregion, R_l, will have rows denoted 1 to N_{R_l}, where N_{R_l} is the number of rows in R_l. The parallelization algorithm is as follows for $l = 1, \ldots, e$ (e is the total number of subregions):

1. In R_l calculate $w^{(n+1)}_{i,j,k}$ for $j = 2, N_{R_l} - 1$ using equations (22) and (23) or (24) and (25) with the corresponding form of (26) at the symmetry boundary for $\Omega(0)$ or D_1, respectively.
2. Pass $w^{(n+1)}_{i,N_{R_l}-1,k}$ from R_l to R_{l+1} except for R_e.
3. Calculate $w^{(n+1)}_{i,1,k}$ in R_{l+1} using data from step (2) and equations (22) and (23) or (24) and (25) with the corresponding form of (26) at the symmetry boundary for $\Omega(0)$ or D_1, respectively.
4. Pass $w^{(n+1)}_{i,1,k}$ from R_{l+1} to R_l except for R_1 to update R_l for the next SOR iteration.

where R_e denotes the last subregion of the division. For the two interface regions $R_{l_I} \in \Omega(0) \cup \Gamma_3$ and $R_{l_{I+1}} \in D_1 \cup \Gamma_3$ between regions $\Omega(0)$ and D_1, steps (1) - (4) of the parallelization algorithm are performed, but the details are more complex, i.e.

1. In R_{l_I} and $R_{l_{I+1}}$ calculate $(w_1)^{(n+1)}_{i,j,k}$ for $j = 2, N_{R_{l_I}}$ and $(w_2)^{(n+1)}_{i,j,k}$ for $j = 2, N_{R_{l_{I+1}}} - 1$ by:
   (a) calculate $(w_1)^{(n+1)}_{i,j,k}$ for $j = 2, N_{R_{l_I}} - 1$ by equations (22), (23), and (26) in $\Omega(0)$.
   (b) calculate $(w_1)^{(n+1)}_{i,N_{R_{l_I}},k}$ for $i = 1, a_1 - 1$ using equation (27) for $\Omega(0)$.
   (c) calculate $(w_1)^{(n+1)}_{i,N_{R_{l_I}},k}$ for $i = a_1$, maxrow1 using equation (32).
   (d) calculate $(w_2)^{(n+1)}_{i,j,k}$ for $j = 2, N_{R_{l_{I+1}}} - 1$ by equations (24), (25), and (26) in D_1.
2. Pass $(w_1)^{(n+1)}_{i,N_{R_{l_I}},k}$ from R_{l_I} to $R_{l_{I+1}}$.

3. Calculate $(w_2)_{i,1,k}^{(n+1)}$ for $i =$ maxcol2, 1 by:
   (a) calculate $(w_2)_{i,1,k}^{(n+1)}$ for $i =$ maxcol2, a_2 by copying the value from $(w_1)_{i,N_{R_{l_I}},k}^{(n+1)}$ and using linear interpolation to fill in the smaller mesh.
   (b) calculate $(w_2)_{i,1,k}^{(n+1)}$ for $i = a_2 - 1, 1$ using equation (27) for D_1.
4. Pass $(w_2)_{i,5,k}^{(n+1)}$ from $R_{l_{I+1}}$ to R_{l_I} for the next SOR iteration.

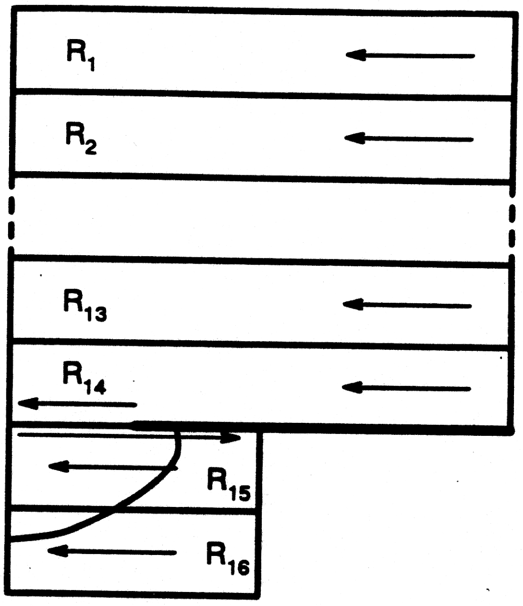

Figure 3: Domain decomposition of mathematical problem into sixteen subregions showing the flow of computations in each.

The advantage of this scheme is that for each iteration, step 1 only depends on data from within the subregion to be calculated and thus this step, which is the bulk of the computation, can be performed in parallel. To maximize the efficiency a load balancing scheme was developed for the particular problem that was solved. This will be discussed in the Results Section.

4.3 Results

In order to compare results with Vuik and Cuvelier [8], we used the same slit opening of $2a = 2.0$ and $B = 10.0$. It was then necessary to find a region $\Omega(0)$ large enough such that if $\Omega(0)$ were enlarged, the etching did not change its shape. We found this to be true experimentally for the serial problem if: $\Delta x = \Delta y = 0.05$, maxrow1 = 280, and maxcol1 = 321 for $\overline{\Omega(0)}$ and $\Delta x = \Delta y = 0.0125$, maxrow2 = 80, and maxcol2 = 161 for $\overline{D_1}$ using a time step $\Delta t = 1.0$, a tolerance of 0.1×10^{-6} for the SOR iterates and 0.1×10^{-3} for the free surface. By a tolerance for the free surface, the solution free surface after convergence at each time step t is defined to be the first value less than the tolerance moving in a direction outward from $\Omega(t)$ in $\overline{D_1}$. It was also found experimentally that $\theta = 1.935$, the SOR relaxation factor, gave the fastest convergence. These parameters fully determine the example problem that was run. Both the serial case and the numerous parallel cases reproduced the results obtained by Vuik and Cuvelier [8] using the finite element method to within approximately 2 gridpoints, i.e. by 0.025 by the twentieth time step, and the results of the serial and parallel cases all agreed with each other exactly except at one point where there was a difference of 0.1×10^{-3}. These differences with Vuik and Cuvelier are probably due to different tolerances and the fact that one method used finite elements and the other used finite differences. Figure 4 shows the moving boundary for $t = 4, 8, 12, 16, 20$ in $\overline{D_1}$ where the etching occurred.

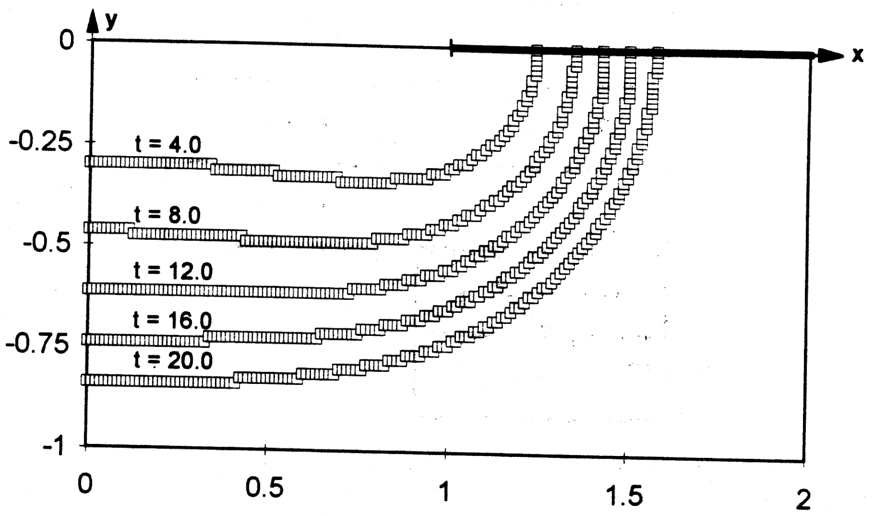

Figure 4: Moving boundary at various times.

Based on the dimensions of the problem the following load balancing scheme was determined, see Table I (and Figure 3), where nodes denotes the number of iPSC/860 CPU's in use and

Table 1: Load Balancing Information for Example

Nodes	2	4	8	16	32	64
bottom nodes	1	1	1	2	4	8
bottom points	12880	12880	12880	6440	3220	1610
top nodes	1	3	7	14	28	56
top points	89880	$\leq$ 30174	12840	6420	3210	1605
diff points	77000	17294	40	20	10	5

how they were divided into the top, $\overline{\Omega(0)}$, and bottom, $\overline{D_1}$, subregions and points denotes the number of gridpoints to be calculated by each node. Note, for 4 nodes we have ≤ 30174

points. This is because 2 nodes ran 29853 points while 1 node ran 30174 points since 280 is not evenly divisible by 3. Hence the problem was load balanced for 8 - 64 nodes.

Figure 5 shows the speedup times for 1-64 nodes at t = 20, i.e. for the 20th time step. Speedup is defined as the ratio of the time to solve the problem on one processor divisible by the time to solve the problem on n processors, where n is the number of nodes. The speedup times reflect the lack of load balancing for the 4 node case which had a significant affect on the efficiency. Similar speedups for the time step at t=4,8,12,16 were achieved. Each node-time combination took 166 ± 6 iterations to run. As can be seen the speedup times were closest to the ideal line (45° line) for 8 nodes, and only started to move off quickly at 64 nodes. This is due to the fact that the communication cost becomes large with respect to the calculation time for each node when the number of rows calculated per node becomes small. One should note that the timing data only took into account the internode communication costs and not the cost of going to and from the host. This is because after the initial condition w_0 is downloaded, no additional communication is necessary with the host because the numerical scheme needs no additional data. The output of the etching profile can be uploaded to the host at the final time step, in our example at $t = 20$. It is clear that if the mesh were refined or the region of calculation were enlarged, the calculation per node would increase and the communication costs for a large number of nodes such as 32 and 64 would become less significant thus improving the speedups. Although the scheme worked well, there does appear to be an ideal number of nodes for the best speedup which is related to the amount of computation performed by each node. This correlation is slightly complicated by the fact that an increase in either the size of the original problem or a mesh refinement would also lead to an increase in the average number of iterations that would increase the communication cost, although more calculation per node and hence communication would be performed at each iteration. The scheme thus appears to parallelize very well on a small scale parallel architecture like the iPSC/860 and would not be well adapted to a large scale parallel architecture such as a Connection Machine.

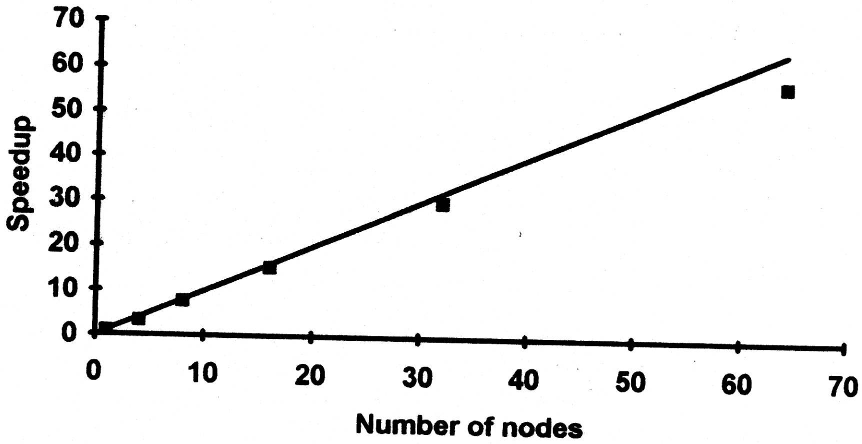

Figure 5: Speedups for various numbers of nodes at t = 20.

5 Acknowledgements

This material is based upon work supported by the National Science Foundation under Award No. ECS-9006516. The authors would also like to thank the San Diego Supercomputer Center for providing time on their Intel iPSC/860 parallel computer and CRAY Y-MP. The authors appreciate Glen Nomi's assistance in obtaining the SOR relaxation factor θ.

References

[1] Crank, J., *Free and Moving Boundary Problems*, Oxford Univ. Press, Oxford, 1984.

[2] Kuiken, H.K., "Etching: A Two-Dimensional Mathematical Approach," *Proc. R. Soc. Lond.*, A392 (1984), pp. 199–225.

[3] Kuiken, H.K., "Etching Through a Slit," *Proc. R. Soc. Lond.*, A396 (1984), pp. 95–117.

[4] Kuiken, H.K., "Mathematical Modeling of Etching Processes," in *Free Boundary Problems: Theory and Applications, I*, Hoffmann, K.H. and J. Sprekels eds., pp. 89–109, Longman Scientific and Technical, Essex, England, 1990.

[5] Kuiken, H.K., J.J. Kelly, and P.H.L. Notten, "Etching Profiles at Resist Edges, I. Mathematical Models for Diffusion-Controlled Cases," *J. Electrochemical Soc.*, 133, No. 6 (1986), pp. 1217–1226.

[6] Lions, J.L., "Introduction to Some Aspects of Free Surface Problems," in *Numerical Solution of Partial Differential Equations III*, Hubbard, B. ed., pp. 376–381, Academic Press, New York, 1976.

[7] Notten, P.H.L., J.J. Kelly, and H.K. Kuiken, "Etching Profiles at Resist Edges, II. Experimental Confirmation of Models Using GaAs," *J. Electrochemical Soc.*, 133, No. 6 (1986), pp. 1226–1232.

[8] Vuik, C. and C. Cuvelier, "Numerical Solution of an Etching Problem," *J. Comput. Phys.*, 59 (1985), pp. 247–263.

[9] Wang, K.P., *Efficient Parallel S.O.R. Iterative Methods for the Solution of a Free Surface Seepage Problem*, pp. 1–130, Ph.D. Thesis, Dept. of Mech. and Envr. Engr., Univ. of Calif., Santa Barbara, CA, 1992.

J.C. Bruch, Jr. and C.A. Papadopoulos
Department of Mechanical and Environmental Engr.
University of California, Santa Barbara, Calif. 93106
J.M. Sloss
Department of Mathematics
University of California, Santa Barbara, Calif. 93106