**Wet Chemical Etching in Semiconductor Fabrication**

**Physical problem** A gap of width 2a and length L is to be etched in a flat plate. The remainder of the plate is covered with a protective (photoresist) layer. Since it is assumed that L is much larger than 2a, the problem can be considered as two dimensional.

![Diagram of wet chemical etching](image)

**Figure 1: Physical Problem.**

**Assumptions:**

- there is no convection in the etching medium;
- the etching process is isotropic;
- the thickness of the photoresist layer is infinitely small;
- only one component of the etching liquid determines the process.
Mathematical Model

The etching fluid $\Omega(t)$ is bounded by the outer boundary $\Gamma_1$; the photoresist layer $\Gamma_2(t)$, and the moving boundary $S(t)$. $\partial\setminus\Omega(t)$ denotes part of the solid.

$$\frac{\partial C}{\partial t} - \tilde{D} \Delta C = 0$$
$$C = C_0 \quad (t = 0)$$

$\Gamma_3$ $\Gamma_2(t) \quad \frac{\partial C}{\partial n} = 0$
$S(t) \quad C = 0$

$\nu_n = -\sigma \frac{\partial C}{\partial n}$
$\tilde{D} \frac{\partial C}{\partial n} = -kC$

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

\[ \frac{\partial w}{\partial t} - \Delta w = 1 \]
\[ w(x, y, 0) = 0 \]
\[ w(x, y, t) > 0 \]

\[ w(x, y, t) = \int_0^t \tilde{C}(x, y, \tau) d\tau \quad (x, y) \in D, \quad t \in (0, T) \]
\[ \tilde{C} = \frac{\partial w}{\partial t} \]

Figure 3: Fixed domain mathematical formulation.
Numerical Algorithm

The basic numerical algorithm is:

\[
(w_1)^{(n+1/2)}_{i,j,k} = \left( \frac{1}{\Delta t} + \frac{2}{(\Delta x)^2} + \frac{2}{(\Delta y)^2} \right)^{-1} \left\{ C_0 + \frac{1}{\Delta t} (w_1)_{i,j,k-1} \\
+ \frac{1}{(\Delta x)^2} \left[ (w_1)^{(n+1)}_{i-1,j,k} + (w_1)^{(n)}_{i+1,j,k} \right] \\
+ \frac{1}{(\Delta y)^2} \left[ (w_1)^{(n+1)}_{i,j-1,k} + (w_1)^{(n)}_{i,j+1,k} \right] \right\}
\]

with

\[
(w_1)^{(n+1)}_{i,j,k} = (w_1)^{(n)}_{i,j,k} + \theta[(w_1)^{(n+1/2)}_{i,j,k} - (w_1)^{(n)}_{i,j,k}]
\]
in \( \Omega(0) \) and

\[
(w_2)^{(n+1/2)}_{i,j,k} = \left( \frac{1}{\Delta t} + \frac{32}{(\Delta x)^2} + \frac{32}{(\Delta y)^2} \right)^{-1} \left\{ -B + \frac{1}{\Delta t} (w_2)_{i,j,k-1} \\
+ \frac{16}{(\Delta x)^2} \left[ (w_2)^{(n+1)}_{i-1,j,k} + (w_2)^{(n)}_{i+1,j,k} \right] \\
+ \frac{16}{(\Delta y)^2} \left[ (w_2)^{(n+1)}_{i,j-1,k} + (w_2)^{(n)}_{i,j+1,k} \right] \right\}
\]

with

\[
(w_2)^{(n+1)}_{i,j,k} = \max\{0, (w_2)^{(n)}_{i,j,k} + \theta[(w_2)^{(n+1/2)}_{i,j,k} - (w_2)^{(n)}_{i,j,k}]\}
\]
in \( D_1 \)
Domain Decomposition

Figure 4: Domain decomposition of mathematical problem into sixteen subregions showing the flow of computations in each.

Load Balancing

Table: Load Balancing Information for Example

<table>
<thead>
<tr>
<th>Nodes</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottom nodes</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>bottom points</td>
<td>12880</td>
<td>12880</td>
<td>12880</td>
<td>6440</td>
<td>3220</td>
<td>1610</td>
</tr>
<tr>
<td>top nodes</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>14</td>
<td>28</td>
<td>56</td>
</tr>
<tr>
<td>top points</td>
<td>89880</td>
<td>≤30174</td>
<td>12840</td>
<td>6420</td>
<td>3210</td>
<td>1605</td>
</tr>
<tr>
<td>diff points</td>
<td>77000</td>
<td>17294</td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>
Speedup

\[ S_p = \frac{T_1}{T_p} \]

where \( T_p \) is the time required to execute the algorithm using \( p \) processors and \( T_1 \) is the time required to execute the same program on a single processor.

Efficiency

\[ e_p = \frac{S_p}{p} \]

Results:

Figure 5: Speedups for various numbers of nodes.
Figure 4: Moving boundary at various times.