

光レジスト現像時の3次元形状シミュレーション

正 員 石塚 辰美<sup>†</sup>

Three-Dimensional Simulation in Photoresist Development

Tatsumi ISHIZUKA<sup>†</sup>, Member

あらまし 半導体集積回路の微細化に伴いデバイス構造を正確に予測するためのシミュレーションが重要になってきている。光を用いたリソグラフィを行う場合、波長による解像力の限界があり、所定のパターンを得るためにはリソグラフィシミュレーションによる最適設計が重要となる。しかしながら、現像時のレジスト表面形状は光の干渉効果による定在波の影響で非常に複雑となり、3次元形状計算が安定的かつ高精度に行える手法が望まれている。本論文では、現像時の3次元形状シミュレーションの新しい手法(ネットワーク法)とその計算結果を報告する。最初に、四面体分割によるネットワークの構成方法を説明する。次に、ネットワーク上の点の移動による現像界面の進行解析手法を示す。この手法の特徴としては、次のようなものがある。界面全体の面情報をもたずにネットワーク上の点だけから現像方向を決定するようにアルゴリズムの工夫がされている。このアルゴリズムはストリングモデルのような特異点(面のループ)消去の必要がなく、安定に複雑な界面形状の計算ができる。最後に、定在波効果が顕著な場合に、安定に3次元現像計算が行えることを示す。

1. ま え が き

半導体集積回路の微細化に伴いデバイス構造のわずかの違いがデバイス特性に大きな影響を及ぼすようになってきた。コンタクト孔形状、配線形状等は特にデバイスの集積度と信頼性にとって重大な要因となっている。デバイス設計を行ううえで、従来にも増して、デバイス形状を正確に予測するためのシミュレーションが重要になってきている。とりわけ、デバイス形状の微細化を支配する技術としてリソグラフィは、活発に研究開発が進められているが、そのなかでも、光リソグラフィは、成熟した技術的安定性と生産性の故に広く使用され、その適用限界を延ばすために多大な努力が払われている。光を用いたリソグラフィを行う場合、波長による解像力の限界があり、所定のパターンを得るためにはリソグラフィシミュレーションによる最適設計が重要となる。2次元形状シミュレーションとしてSAMPLE<sup>(1)</sup>等が広く利用されているが、集積化の要求による設計線幅の縮小とデバイス形状の複雑化により現像時の3次元効果が無視できなくなってい

る<sup>(2)</sup>。しかしながら、現像時のレジスト形状は光の干渉効果による定在波の影響で非常に複雑となり、3次元形状計算が安定的かつ高精度に行える手法が望まれている。

3次元リソグラフィシミュレータとしては既に種々の報告<sup>(2)-(4),(7)-(9)</sup>がされている。これらで用いられている現像形状計算の手法としては、ストリングモデル<sup>(8)</sup>、レイトレーシングモデル<sup>(2),(6)</sup>、セルモデル<sup>(7)</sup>、修正拡散モデル<sup>(9)</sup>などがある。ストリングモデルを用いて3次元現像計算を行うためには面の移動に伴う複雑な幾何学処理が必要となる。レイトレーシングモデルを用いる場合も定在波効果の顕著な場合には精度を上げるためにストリングモデルと同じ処理に基づいて面の計算を行い計算点の増減を行わなければならない<sup>(2)</sup>。修正拡散モデルは、拡散方程式を用いる斬新な手法である。本論文で述べるネットワーク法は、3次元空間に張られた網目(ネットワーク)上の点の移動により現像界面の進行を表す手法であり、従来の手法と比較して単純なアルゴリズムであり、かつ安定的に計算が行える点で極めて実用的である。この手法の本質的部分は3次元空間の自由曲面の移動を1次元の計算だけで処理するところにある。

本論文では、3次元形状シミュレーションの新しい

<sup>†</sup> (株)富士総合研究所, 東京都  
Fuji Research Institute Corporation, Tokyo, 108 Japan

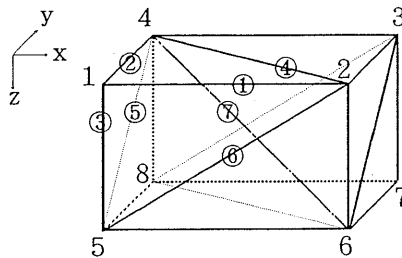
手法；ネットワーク法の紹介とネットワーク法を用いた光レジスト現像シミュレーション結果の報告を行う。

## 2. 現像形状計算手法

リソグラフィシミュレーションは Dill らの方法<sup>(5)</sup>に従えば、①レジスト表面の光強度計算、②レジスト内の露光計算、③現像計算の3部分より構成される。現像計算について考えると、まず露光量から溶解(現像)速度を計算し、次に現像形状計算をする。ここで報告するネットワーク法は、現像形状計算の新しい手法であり、3次元計算を安定的かつ高精度に行うために開発した。ネットワーク法はレジスト内の3次元空間に張られたネットワーク上の点の移動により現像界面の形状変化を取り扱う手法である。以下にネットワーク法の計算手法の説明を行う。

### 2.1 ネットワークの定義

レジストの計算領域は、直方体とする。現像されるレジストを直交メッシュにより分割する。これにより



- 1:  $P_{i j k}$       ①:  $L_{ijk1}$
- 2:  $P_{i+1 j k}$     ②:  $L_{ijk2}$
- 3:  $P_{i+1 j+1 k}$    ③:  $L_{ijk3}$
- 4:  $P_{i j+1 k}$       ④:  $L_{ijk4}$
- 5:  $P_{i j k+1}$       ⑤:  $L_{ijk5}$
- 6:  $P_{i+1 j k+1}$    ⑥:  $L_{ijk6}$
- 7:  $P_{i+1 j+1 k+1}$  ⑦:  $L_{ijk7}$
- 8:  $P_{i j+1 k+1}$

図1 ユニットの頂点と稜線番号  
数字はユニット  $U_{ijk}$  内の頂点1~8と稜線①~⑦を示す  
Fig. 1. Relations of vertices and edges in the unit  $U_{ijk}$ .  
The numbers 1 to 8 are the vertices, and ① to ⑦ the edges in the unit  $U_{ijk}$ .

できた微小直方体をユニットと呼ぶ。更にこのユニットを図1に示すように6個の四面体に再分割する。

この四面体をセルと呼ぶ。セルの頂点と頂点間を結ぶ稜線(有向線分)をネットワーク  $\Gamma$  と呼ぶ。図1にはユニットの頂点とセルの稜線の関係も示されている。ネットワーク法で用いる幾何要素を表1に示す。セル  $C_{ijkl}$  と頂点  $P_{ijk}$  の関係、セル稜線  $L_{ijkm}$  と頂点  $P_{ijk}$  の関係は以下のとおりである。

$$C_{ijk1} = P_{ijk}P_{i+1jk}P_{ij+1k}P_{ij+1k+1} \quad (1a)$$

$$C_{ijk2} = P_{i+1jk}P_{ij+1k}P_{ij+1k+1}P_{i+1jk+1} \quad (1b)$$

$$C_{ijk3} = P_{ij+1k}P_{ijk+1}P_{i+1jk+1}P_{ij+1k+1} \quad (1c)$$

$$C_{ijk4} = P_{i+1jk}P_{i+1j+1k}P_{ij+1k}P_{i+1jk+1} \quad (1d)$$

$$C_{ijk5} = P_{i+1j+1k}P_{ij+1k}P_{i+1jk+1}P_{ij+1k+1} \quad (1e)$$

$$C_{ijk6} = P_{i+1j+1k}P_{i+1jk+1}P_{i+1j+1k+1}P_{ij+1k+1} \quad (1f)$$

$$L_{ijk1} = P_{ijk}P_{i+1jk} \quad (2a)$$

$$L_{ijk2} = P_{ijk}P_{ij+1k} \quad (2b)$$

$$L_{ijk3} = P_{ijk}P_{ijk+1} \quad (2c)$$

$$L_{ijk4} = P_{i+1jk}P_{ij+1k} \quad (2d)$$

$$L_{ijk5} = P_{ij+1k}P_{ijk+1} \quad (2e)$$

$$L_{ijk6} = P_{i+1jk}P_{ijk+1} \quad (2f)$$

$$L_{ijk7} = P_{i+1jk}P_{i+1j+1k} \quad (2g)$$

各稜線の向きは式(2a)~(2g)の定義のとおりとする。図2にユニット内のセル、稜線、頂点の関係を示す。

形状計算をするうえでの基本単位は、セルであるが、解析領域をセル分割するための補助手段としてユニットを用いている。また、形状計算を進めるうえで3次元空間内の四面体のつながりは複雑となるので、計算の補助的目的でユニットを用いた。

### 2.2 現像界面と現像計算データ

現像界面はネットワーク上の点  $Q(t)$  (界面定義点と呼ぶ)の集合として次式で表現される。

$$Q_t = \{Q(t); Q(t) \in L_{ijkm}, t \text{ は現像時間}\} \quad (3)$$

但し、一つの稜線上には唯一の界面定義点しかもたないとする。現像速度は各頂点で定義されている。各稜線上の任意点での現像速度は稜線両端間の線形補間に

表1 ネットワーク法で用いる幾何要素

集合名	幾何要素
ユニットの集合	$\Psi = \{U_{ijk} : \text{ユニット}, i=1 \sim m_1, j=1 \sim m_2, k=1 \sim m_3\}$
セルの集合	$\Omega = \{C_{ijkl} : \text{セル}, l=1 \sim 6, C_{ijkl} \subseteq U_{ijk}\}$
頂点の集合	$V_t = \{P_{ijk} : \text{頂点}, i=1 \sim m_1+1, j=1 \sim m_2+1, k=1 \sim m_3+1\}$
稜線の集合	$E_\sigma = \{L_{ijkm} : \text{セルの稜線}, m=1 \sim 7\}$
セル面の集合	$\Sigma = \{F_{ijk\nu} : \text{セルの面}, \nu=1 \sim 12\}$
界面定義点の集合	$Q_t = \{Q(t) : Q(t) \in L_{ijkm}, t : \text{現像時間}\}$
ネットワーク	$\Gamma = V_t \cup E_\sigma$

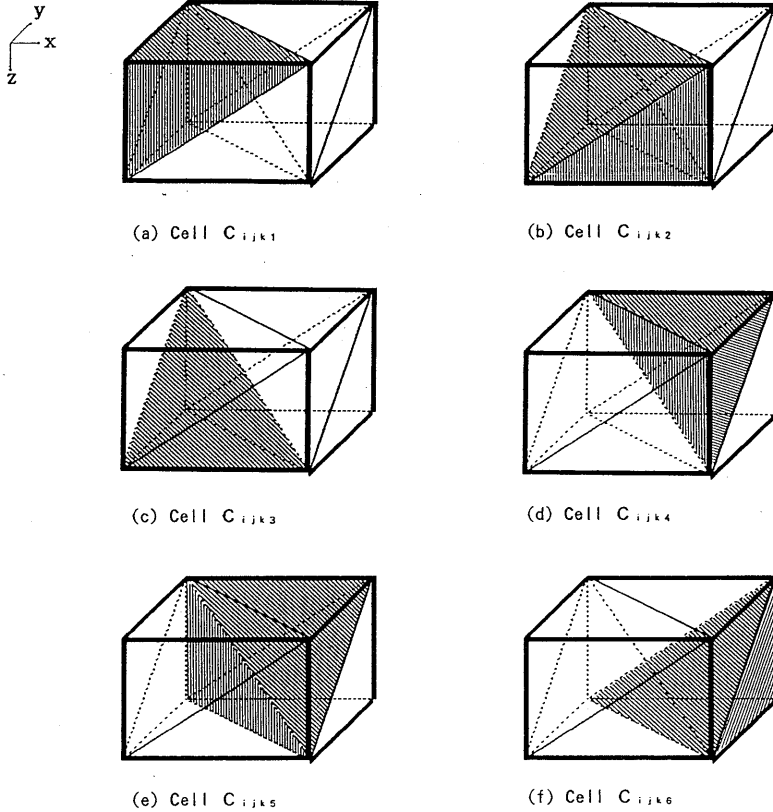


図2 ユニット内のセルと稜線と頂点の関係  
Fig. 2 Cell decompositions of the unit  $U_{ijk}$ .

より求める。稜線上の現像時間、現像距離の計算方法は付録1.に示す。  $Q(t)$  は付録1.に示すような現像速度に従って稜線  $L_{ijkm}$  上を進むものとする。

現像計算中には、頂点で保持すべきデータと稜線ごともつデータに分かれる。頂点ごとに定義されるデータは現像速度  $v(P)$  と現像フラグ  $I(P)$  である。稜線ごとに定義されるデータは現像方向フラグ  $K(L)$ 、現像方向ベクトル  $W(Q(t))$ 、稜線ベクトル  $E(L)$ 、稜線上の現像界面位置  $A(L)$  である。現像フラグ  $I(P)$  は0,1の値で現像界面が到着するまでは1となっていて、それ以後は0となる。従って、初期状態では表面のすべての頂点は0、表面以外の頂点は1となっている。式(2a)~(2g)で決めた稜線の向きに従って稜線  $L$  の始点と終点を  $P_1, P_2$  とすると現像方向フラグは次のように定義される。

$$K(L) = I(P_2) - I(P_1) \quad (= -1, 0, 1) \quad (4)$$

$K(L)$  は稜線の向きと現像進行方向のが同一のときは1となり、逆のときは-1となる。0のときはその稜線

に現像界面が存在しない場合である。式(5)で定義する稜線ベクトル  $E(L)$  は長さが稜線に等しく向きが現像進行方向である。

$$E(L) = K(L) \cdot (P_2 - P_1) \quad (5)$$

但し  $P_1, P_2$  は稜線  $L$  の始点、終点の位置ベクトルである。現像界面位置  $A(L)$  は稜線  $L$  上の界面位置のデータで0~1の値をもつ。1のときは現像されていない状態で0のときは現像が終了した状態を表す。

$$A(L) = 1 - (L \text{ の現像済み長さ}) / (L \text{ の長さ}) \quad (6)$$

現像方向ベクトル  $W(L)$  は界面定義点  $Q(t)$  の現像方向を表す。現像方向の決定方法は次の節で述べる。

### 2.3 現像方向と稜線上の現像速度

現像方向を決定するアルゴリズムは形状計算の精度を保証するために重要である。直交メッシュにより決まるユニットを四面体のセルに再分割したのはこの点を重視したことによる。同時に四面体の各面が三角形を構成し現像界面との交線が直線として一意に定められることも四面体を採用した理由である。

稜線  $L$  上の点  $Q(t)$  の現像速度は現像界面の法線方向と稜線の方向の関係を用いて決める。  $Q(t)$  における現像界面の法線方向は  $Q(t)$  につながるすべての現像界面の平均的法線方向として決定する。 そのために、稜線  $L$  を共有しているすべてのセルの面(三角形)上で現像界面との交線(界面線ベクトル  $W_F$  と呼ぶ)を以下のように定義する。

$$W_F = P_A - P_B \quad (7)$$

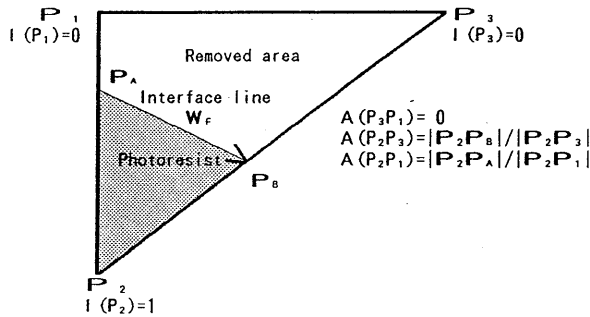
$P_A, P_B$  は、稜線上の界面定義点で、両端の  $I(P)$  が異なる場合に定義される点である。 この関係を図 3 (a) に

示す。 現像界面をもたない面では  $W_F$  は零となる。 また、稜線とそれを共有するセル面の関係は付録 2. に示す。 稜線  $L$  を共有するすべてのセル面上の界面線ベクトル  $W_{Fi}$  を求め、これらを用いて現像方向ベクトル  $W(Q(t))$  は次のように計算する。

界面線ベクトル相互の外積  $W'_{ij}$  を稜線の現像方向と同じ向きとなるように符号を決める。 次にこれらの和を求め、これを規格化し、現像方向  $W(Q(t))$  を決める。

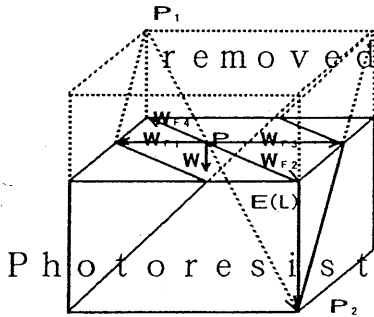
$$W'_{ij} = (W_{Fi} \times W_{Fj}) \quad (8a)$$

$$W_{ij} = \text{sign}(1, (E(L) \cdot W'_{ij})) \cdot W'_{ij} \quad (8b)$$

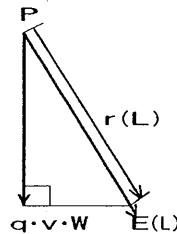


$P_i$ : Position vector of the vertex  $P_i$   
 $I(P_i)$ : Development flag  
 $A(P_iP_j)$ : Development fraction of the edge line  $P_iP_j$

(a) Interface vector on the face of the tetrahedral cell.



(b) Development direction  $W$  at  $P$  calculated by interface vectors  $W_{Fi}$ .



(c) Development rate  $r(L)$  at  $P$  evaluated on the vector  $E(L)$ .

$W_{i,j} = W_{Fi} \times W_{Fj}$ : normal vector of the surface spanned by  $W_{Fi}$  and  $W_{Fj}$   
 where  $ij=12, 23, 34, 41$

$W = \sum W_{i,j} / |\sum W_{i,j}|$ : development direction at  $P$

$v = v(P_2) + \{v(P_1) - v(P_2)\} \cdot A(L)$ : development rate of  $P=Q(t)$

$v(P_i)$ : development rate of edge point  $P_i$

$q$ : parameter of fineness

図 3 現像界面線と現像方向

Fig. 3 Interface vectors and development direction.

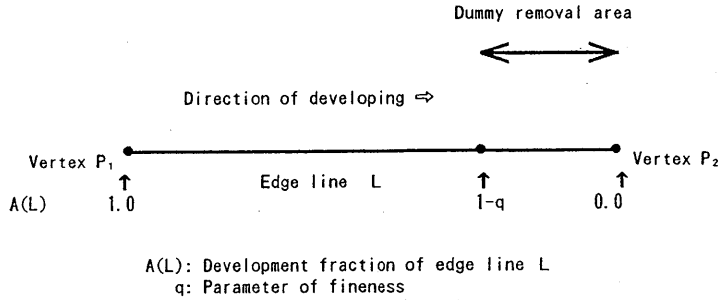


図4 稜線上の現像終了見なし領域  
Fig. 4 Dummy removal area.

$$W(Q(t)) = \sum W_{ij} / |\sum W_{ij}| \quad (8c)$$

ここで  $\text{sign}(a, b)$  は  $a, b$  が同符号の場合に 1 となり異符号の場合 -1 となる。  $W_{Fi}$  と  $W$  の関係を図 3 (b) に示す。  $Q(t)$  点の現像方向は単位ベクトル  $W$  で表す。  $Q(t)$  点の現像速度  $v = v(Q(t))$  は稜線の両端点の現像速度から補間して求める。 界面定義点  $Q(t)$  は稜線  $L$  上しか動かないので、実際の現像方向  $W$  と  $E(L)$  を用いて次のように稜線  $L$  上の現像速度  $r(L)$  を求める。

$$r(L) = q \cdot [v(P_2) + \{v(P_1) - v(P_2)\} \cdot A(L)] \cdot |E| \cdot |W| / (E \cdot W) \quad (9)$$

ここで  $q$  は次節で説明する精度パラメータ、  $v(P_i)$  は稜線端点の現像速度である。 この関係を図 3 (c) に示す。 また、境界条件として、計算領域の境界上にある稜線では現像方向  $W$  が  $XY$ -面に垂直としている。

ここに示したようなデータをもつことの意味はすべての処理を稜線上だけで行い、一つの稜線上に二つの界面をもたないようにすることである。 セル (四面体) の面 (三角形) 上の唯一の線が現像界面を表すようにすると界面定義点のみで面の構成が決まる。 このようにすることにより自動的に特異点のない滑らかな面を得ることができる。 しかも四面体を用いているので形状表現能力に優れている。

一般に、セルモデルを用いた場合は、セルの界面が必ず現像界面となり、実際の現像界面とずれが生じる。 現像は表面反応としているので、表面積が現像量を支配している。 従って、セルモデルのようにセルの界面と実際の現像界面の間にずれがある場合には、表面積の違いを現像速度に換算し、現像速度の補正を行わなければならない<sup>(7)</sup>。 ネットワーク法は、セルの稜線上の界面定義点を使うことにより、現像界面をセルごとではなく連続的に定義することができる。 従ってセルモデルで用いるような現像速度の補正は必要がない。

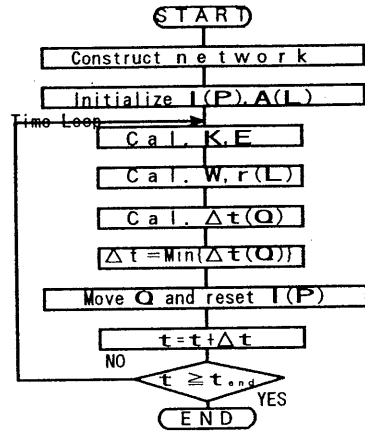


図5 3次元現像形状計算の流れ図  
Fig. 5 Flow chart of topography calculation.

2.4 タイムステップコントロールと精度パラメータ  
形状計算は、現像時間を微小時間 (タイムステップ) に区切ってタイムステップごとに稜線上の界面定義点の進行を計算する。 タイムステップは 1 回の計算で  $Q(t)$  が一つの稜線を越えないように自動的に制御する。 つまり、各稜線上の  $Q(t)$  が稜線の端に到達する時間を  $\Delta t(Q(t))$  を求め、すべての稜線についての最小値を  $\Delta t$  とする。  $\Delta t$  を 1 回の現像タイムステップとする。  $\Delta t$  後の各稜線上の現像界面位置を  $A(L)$  として、

$$A(L) \leq (1-q) \quad (q < 1) \quad (10)$$

が満足された場合、  $I(P_2) = 0$  とする。 このとき、稜線  $L$  の現像が終了したとする。 この関係を図 4 に示す。  $q$  を精度パラメータと呼ぶ。  $q$  が小さいとセルリムーバの処理に近くなる。  $q$  が 1.0 に近づくと飛躍的に計算時間が増える。 3.2 で精度パラメータと計算時間の関係を示す。 実用上はメッシュを細かくして  $q = 0.9$  位にするのが望ましい。

2.5 現像形状計算のまとめ

ここでは、現像形状計算のアルゴリズムを整理して示す。図5には形状計算の概要を流れ図にして示す。形状計算の処理手順は、(1)ネットワークの構成、(2)現像フラグ  $I(P)$ 、現像界面位置  $A(L)$  の初期化、(3)現像方向フラグ  $K(L)$ 、稜線ベクトル  $E(L)$  の計算、(4)現像方向ベクトル  $W$ 、稜線上の現像速度  $r$  の計算、(5)稜線上での端点への現像到達時間の計算、(6)

タイムステップの決定、(7)界面定義点の更新と現像フラグの更新、(8)時刻の更新、(9)終了判定となる。(3)から(9)がタイムステップのループとなる。

3. 計算例

計算例として定在波効果の顕著な場合の3次元現像計算の結果を示す。また、同じ条件でメッシュと精度パラメータを変えた計算を行い、計算時間がメッシュ

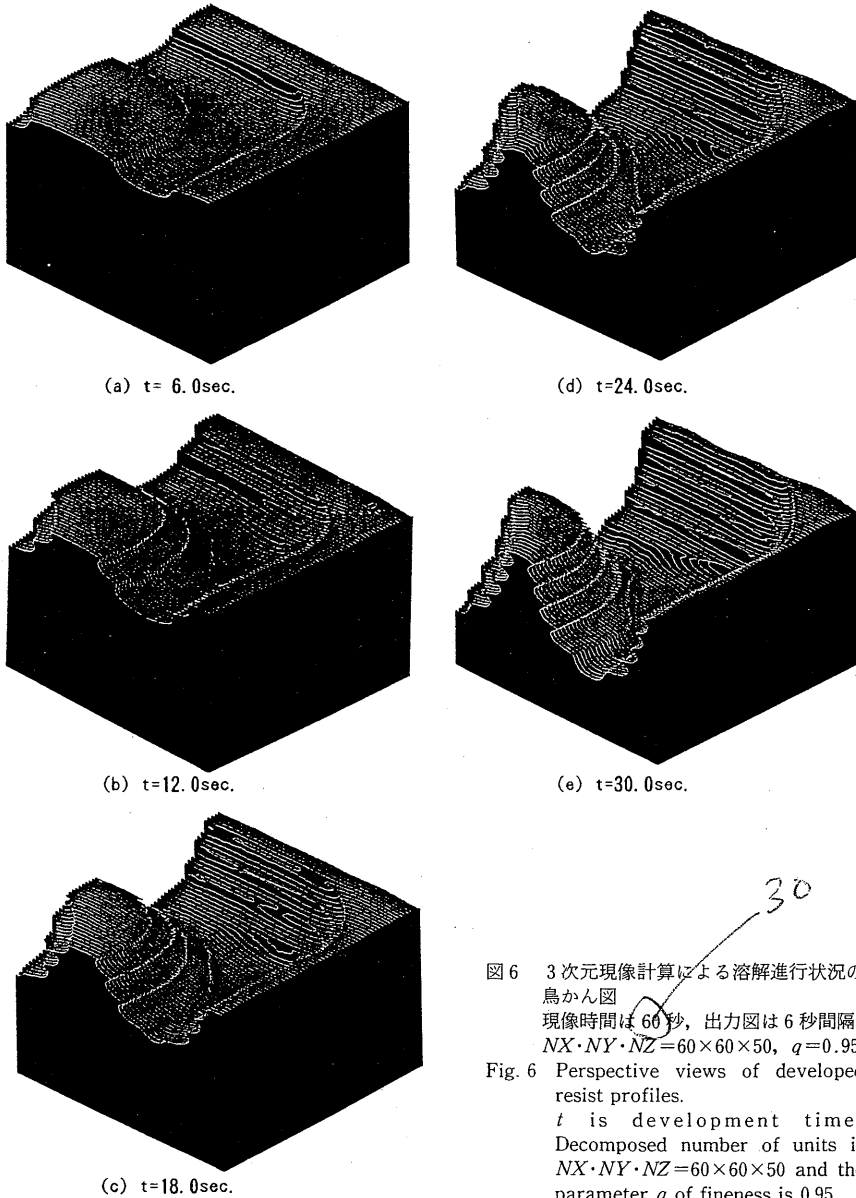


図6 3次元現像計算による溶解進行状況の鳥かん図  
 現像時間は60秒、出力図は6秒間隔、  
 $NX \cdot NY \cdot NZ = 60 \times 60 \times 50$ ,  $q = 0.95$   
 Fig. 6 Perspective views of developed resist profiles.  
 $t$  is development time.  
 Decomposed number of units is  $NX \cdot NY \cdot NZ = 60 \times 60 \times 50$  and the parameter  $q$  of fineness is 0.95.

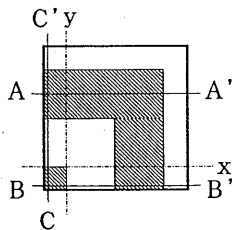
数と精度パラメータにどのように依存性するかを調べた。

### 3.1 3次元現像計算結果

定在波効果が顕著な場合の3次元現像計算結果を図6に示す。図6の計算は、 $x$ ,  $y$ ,  $z$ 方向のユニット分割数(図の記号で  $NX$ ,  $NY$ ,  $NZ$ )としてそれぞれ60, 60, 50の等分割を行っている。セル(四面体)の数としては、1080000個である。精度パラメータ  $q$  として0.95を用いた。現像条件としては、Moniwaらの報告<sup>(2)</sup>と同じものである。この条件を表2に示す。現像に用いたマスクは、図7のようなL字形の0.5  $\mu\text{m}$ ライン&スペースである。A-A'の現像断面図(図8(a))は、Moniwaらの報告<sup>(2)</sup>と同様の結果となっている。光強の影響で角の部分が深くなる傾向は図8(a)に、マス

表2 光レジスト現像条件

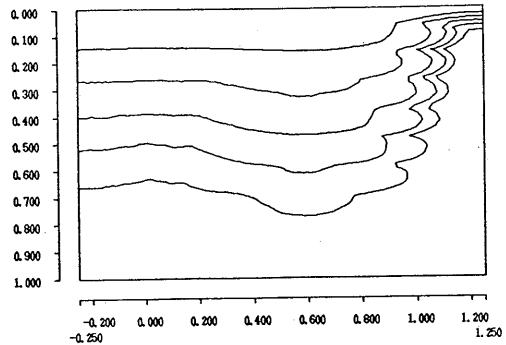
光学条件	光波長	0.365 $\mu\text{m}$
	N. A.	0.42
	デフォーカス	1.0 $\mu\text{m}$
	コヒーレンシー	0.5
	ドーズ量	72.0 $\text{mJ}/\text{cm}^2$
レジスト光学条件	A	0.52/ $\mu\text{m}$
パラメータ	B	0.062/ $\mu\text{m}$
	C	0.018 $\text{cm}^2/\text{mJ}$
レジスト膜厚		1.0 $\mu\text{m}$
レジスト屈折率		1.72
基板複素屈折率		(7.0, -2.16)
現像速度	$E_1$	8.09
パラメータ	$E_2$	-5.32
	$E_3$	-0.01
現像時間		30.0秒



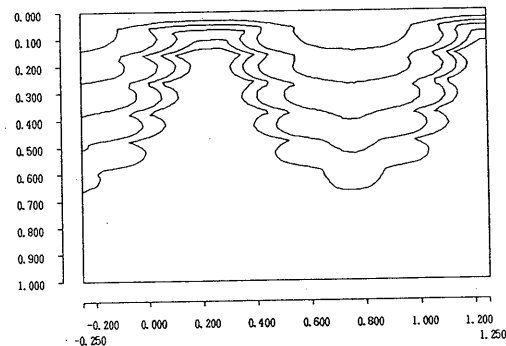
hatched parts: complete transparent  
background : perfect opaque  
size of mask:  $x=1.5\mu\text{m}(x=-0.25\sim 1.25\mu\text{m}), y=1.5\mu\text{m}(y=-0.25\sim 1.25\mu\text{m})$   
line A-A': cross section of  $y=0.75\mu\text{m}$   
line B-B': cross section of  $y=-0.2\mu\text{m}$   
line C-C': cross section of  $x=-0.2\mu\text{m}$

図7 L字形0.5  $\mu\text{m}$ ライン&スペースのマスク  
Fig. 7 Mask pattern of 0.5- $\mu\text{m}$ -wide line and space.

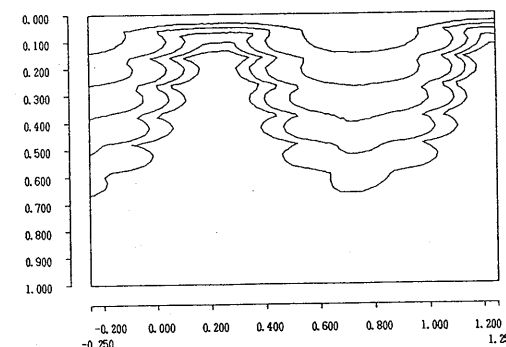
クの角の部分が円く現像される傾向は、図6に良く現れている。また、顕著な定在波効果を3次元形状として良く表現している。図8(b), (c)にB-B'断面とC-C'断面の比較を示す。二つの断面図はほとんど同じ形状を示している。これはユニットのセル分割が  $x$  方



(a) Developed resist profiles at cross-section A-A' ( $y=0.75 \mu\text{m}$ )



(b) Developed resist profiles at cross-section B-B' ( $y=-0.2 \mu\text{m}$ )



(c) Developed resist profiles at cross-section C-C' ( $x=-0.2 \mu\text{m}$ )

図8 異なる断面の溶解進行図

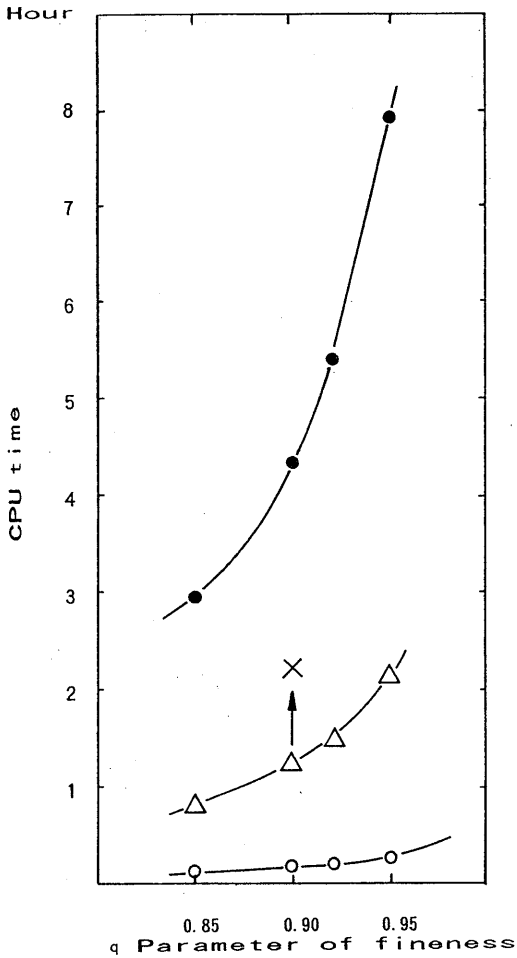
$$NX \cdot NY \cdot NZ = 60 \times 60 \times 50, q = 0.95$$

Fig. 8 Developed resist profiles at different cross-sections after 6 s, 12 s, 18 s, 24 s and 30 s of development time. ( $NX \cdot NY \cdot NZ = 60 \times 60 \times 50, q = 0.95$ )

向と  $y$  方向で異なるにもかかわらず、良い精度で形状計算が行われていることを示している。このように、ネットワーク法では面のループなど特異点処理を行うことなく精度良く安定に形状計算ができる。

3.2 計算時間のメッシュ数と精度パラメータ依存性

メッシュ数と精度パラメータを変えた場合の計算時間の変化を図9に示す。図9に示した計算時間はCONVEX-C210を使用したCPU時間である。ここに示した計算では、スカラー計算は、最適化レベル-01であり、それ以外はすべて最適化レベル-02のベクトル化を



- : cases of  $NX \cdot NY \cdot NZ = 60 \times 60 \times 50$
- △ : cases of  $NX \cdot NY \cdot NZ = 40 \times 40 \times 40$
- : cases of  $NX \cdot NY \cdot NZ = 20 \times 20 \times 30$
- X : case of scalar mode calculation,  $NX \cdot NY \cdot NZ = 40 \times 40 \times 40$  and  $q = 0.9$

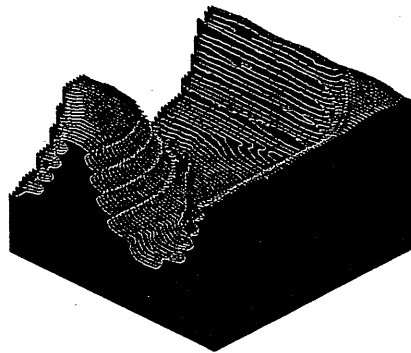
図9 計算時間のメッシュ数と精度パラメータ依存性  
Fig. 9 Variation in CPU time as number of units and the parameter of fineness.

行っている。図9に示した計算はメッシュ数と精度パラメータ以外はすべて3.1の条件と同じである。

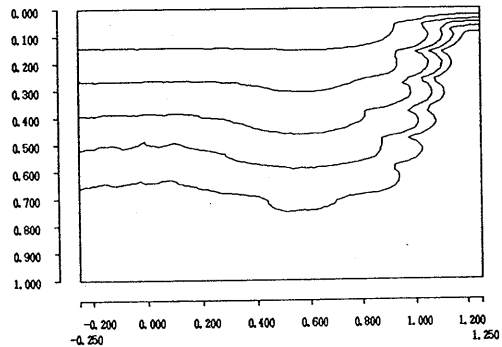
図9からわかるように、 $q$ が1に近づくと計算時間が急激に増加する。しかもメッシュ数が増えると計算時間の増加割合がより激しくなる。これは1回のタイムステップで界面定義点が一つの稜線を越えて進まないように制御していることによる。

精度パラメータが形状に与える影響を試みる。メッシュ数が多い場合 ( $NX \cdot NY \cdot NZ = 60 \times 60 \times 50$ ) には図8(a)と図10(b)を比較してわかるように、 $q = 0.85$ と $q = 0.95$ の間に計算した形状に大差がない。

次に、メッシュ数による形状の違いを示す。図6と図11(a), (b)の比較、図8(a)と図11の(c), (d)の形状の比較を見ると、メッシュ数が多い場合には、より滑らかな形状がえられるが、メッシュ数が極端に



(a) Perspective view ( $t = 30$  s)



(b) Developed resist profiles of (a) at cross-section A-A' ( $y = 0.75 \mu\text{m}$ ) after 6 s, 12 s, 18 s, 24 s and 30 s of development time

図10 精度パラメータを変えた場合 ( $q = 0.85$ ) の溶解進行図  
 $NX \cdot NY \cdot NZ = 60 \times 60 \times 50$

Fig. 10 Developed resist profiles as the parameter  $q$  of fineness is 0.85.  
( $NX \cdot NY \cdot NZ = 60 \times 60 \times 50$ )



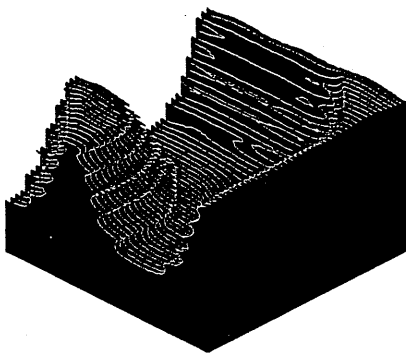
少ない場合 ( $NX \cdot NY \cdot NZ = 20 \times 20 \times 30$ ) にも定性的に同様の計算結果となる。このように、メッシュ数に対する依存性が大きくなく、メッシュ数が少なくとも定性的に同じ傾向の形状計算結果を得られるという点で、ネットワーク法は実用性に富む手法と言える。

図9で、1例だけスカラ計算とベクトル化計算の比

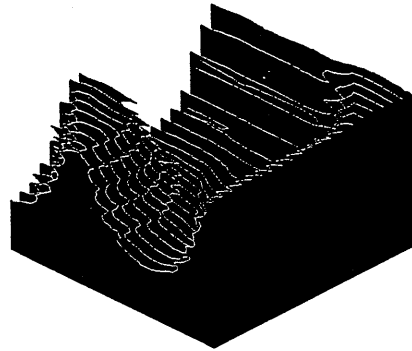
較を行っている。この例ではベクトル化により2倍の高速化が実現されている。

#### 4. むすび

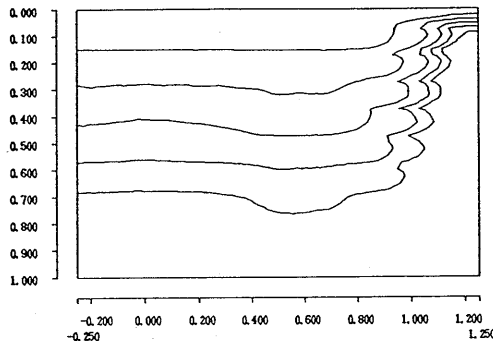
3次元形状計算の新しいアルゴリズム；ネットワーク法を紹介し計算例を示した。その中でネットワーク



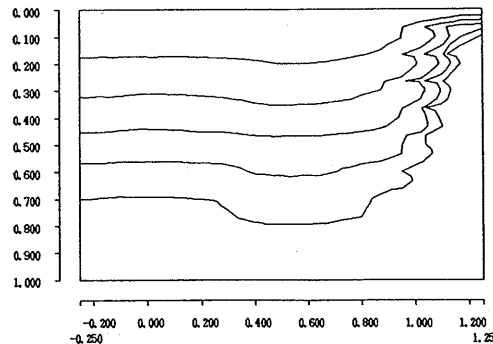
(a) Perspective view  
 $NX \cdot NY \cdot NZ = 40 \times 40 \times 40$ ,  $q = 0.95$   
 $t = 30 \text{ sec.}$



(b) Perspective view  
 $NX \cdot NY \cdot NZ = 20 \times 20 \times 30$ ,  $q = 0.95$   
 $t = 30 \text{ sec.}$



(c) Developed resist profiles of (a) at cross-section A-A' ( $y = 0.75 \mu\text{m}$ ) after 6 s, 12 s, 18 s, 24 s and 30 s of development time.



(d) Developed resist profiles of (b) at cross-section A-A' ( $y = 0.75 \mu\text{m}$ ) after 6 s, 12 s, 18 s, 24 s and 30 s of development time.

図11 メッシュ数を変えた場合の溶解進行比較図

Fig. 11 Comparison between developed resist profiles by  $NX \cdot NY \cdot NZ = 40 \times 40 \times 40$  and  $NX \cdot NY \cdot NZ = 20 \times 20 \times 30$ .

法の計算手法として特徴的なデータ構造について詳しく説明し、この方法の次のような特徴を明らかにした。すなわち、(1)四面体を用いたネットワークの構成、(2)ネットワーク上の点の移動により現像界面の変化を追跡する、(3)四面体の面と現像界面の交線を唯一定義する、(4)これらの交線を用いて現像方向を決定する、(5)精度を保証するようにタイムステップを制御する、などである。このような処理を行うことにより界面全体の面のデータをもたずに計算を行うことが可能となり、特異点の処理が不要となる。従って、定在波効果が顕著な場合の複雑な形状も安定的に計算できることを示した。メッシュ数と精度パラメータを変えた場合の計算時間依存性と計算結果の比較を行い、ネットワーク法が実用的であることを示した。

今後の課題としては、光リソグラフィでの応用例を広げると共に、光以外のリソグラフィに適用することである。また、ベクトル化処理の効率を上げて更に高速化を図ることである。

謝辞 本論文は、ソニー株式会社の依頼により開発した3次元リソグラフィシミュレータ“TREPTON”の開発成果の一部である。出力例については、“TREPTON”を使用した。本論文を公表することを許可されたソニー株式会社の理学博士八木厚夫氏<sup>†</sup>、木村光紀氏に感謝致します。本論文をまとめるに当って御指導頂いた広島大学集積化システム研究センター長、工学博士廣瀬全孝教授、同センター工学博士小柳光正教授に感謝致します。

## 文 献

- (1) W. Oldham, S. Nandgaonker, A. Neureuther and M. O'Toole: "A general simulator for VLSI lithography and etching processes: part-I Applications to projection lithography", IEEE Trans. Electron Devices, **ED-26**, 4, pp. 717-722 (1979).
- (2) A. Moniwa, T. Matsuzawa, T. Ito and H. Sunami: "A three-dimensional photoresist imaging process simulator for strong standing-wave effect environment", IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst., **CAD-6**, 3, pp. 431-438 (1987).
- (3) F. Jones and J. Paraszczak: "RD3D (Computer simulation of resist development in three dimensions)", IEEE Trans. Electron Devices, **ED-28**, 12, pp. 1544-1552 (1981).
- (4) H. Berz, K. Heinrich, A. Heuberger, H. Huber and H. Oertel: "Resolution limits in x-ray lithography calculated by means of x-ray lithography simulator

- Xmas", J. Vac. Sci. Technol., **B4**(1) (1986).
- (5) F. H. Dill, A. R. Neureuther, J. A. Tuttle and E. J. Walker: "Modeling projection printing of positive photoresists", IEEE Trans. Electron Devices, **ED-22**, 7, pp. 456-464 (1975).
- (6) P. I. Hagouel: "X-ray lithographic fabrication of blazed diffraction grating", Ph. D. Dissertation, Univ. of California, Berkeley (1976).
- (7) 平井義彦, 笹子 勝, 遠藤政孝, 野村 登, 羽山 繁: "セル・モデルを用いた3次元 Photo, EB リソグラフィ・シミュレータ PEACE の開発", 信学技報, **SSD86-80** (1986-09).
- (8) 伊藤, 松澤, 門田, 花島: "レジスト3次元形状シミュレータおよびそのサブミクロンプロセスへの応用", 信学論(C), **J70-C**, 3 (昭 62-03).
- (9) M. Fujinaga, N. Kotani, H. Oda, M. Shirahata, H. Genjo, T. Katayama, T. Ogawa and Y. Akasaka: "Three dimensional topography simulation model using diffusion equation", Technical digest of IEDM'88, p. 332 (1988).

## 付 録

### 1. 現像速度こう配が一樣な媒質中の現像速度

ネットワーク法で現像シミュレーションを行うためには、現像速度のこう配が一樣な媒質中の現像速度を求める必要がある。

1次元空間で図A・1に示すように、点  $P_a$  と点  $P_b$  を考え、各々の点での現像速度を  $a$ ,  $b$  とする。  $L$  は、  $P_a$  と  $P_b$  の距離とする。  $P_a$  から  $P_b$  までの現像時間を  $s$  とし、現像速度、現像到達位置を時間の関数として  $v(t)$ ,  $x(t)$  と表す。ここでは、  $v(t)$ ,  $x(t)$ ,  $s$  を  $a$ ,  $b$ ,  $L$  より求める。

次の関係が成立している。

$$dv/dx = \beta \quad (\text{A} \cdot 1)$$

但し、  $\beta = (b - a)/L$

式(A・1)は、次のように変形される。

$$dv/dt - \beta dx/dt = 0 \quad (\text{A} \cdot 2)$$

この式を解くと、

$$v(t) = Ae^{\beta t} + C \quad (\text{A} \cdot 3)$$

となる。

境界条件と距離関係は以下のようなになる。

$$v(0) = a \quad (\text{A} \cdot 4)$$

$$v(s) = b \quad (\text{A} \cdot 5)$$

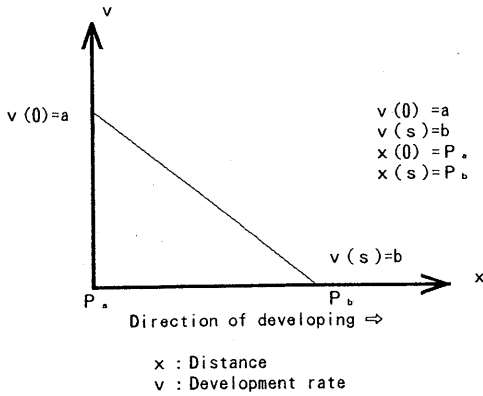
$$\int_0^s v(t) dt = L \quad (\text{A} \cdot 6)$$

式(A・3)を式(A・4)~(A・6)の条件で解くと、

$$v(t) = a \cdot e^{\beta t} \quad (\text{A} \cdot 7)$$

となる。

<sup>†</sup> 現在は日本プレジジョン・サーキット株式会社に勤務されています。



図A.1 1次元空間での現像速度

Fig. A.1 Development rate in one-dimensional space.

表 A.1 稜線とセル面の関係

$\nu$ (セル面番号)	面 構 成 稜 線		
1	$L_{ijk1}$	$L_{ijk2}$	$L_{ijk4}$
2	$L_{ijk1}$	$L_{ijk3}$	$L_{ijk6}$
3	$L_{ijk2}$	$L_{ijk3}$	$L_{ijk5}$
4	$L_{ijk4}$	$L_{ijk5}$	$L_{ijk6}$
5	$L_{i+1jk3}$	$L_{ijk6}$	$L_{ijk+1 1}$
6	$L_{ijk5}$	$L_{ijk7}$	$L_{ijk+1 1}$
7	$L_{ijk5}$	$L_{ij+1k3}$	$L_{ijk+1 2}$
8	$L_{i+1jk2}$	$L_{ij+1k1}$	$L_{ijk4}$
9	$L_{i+1jk3}$	$L_{ijk4}$	$L_{ijk7}$
10	$L_{i+1jk5}$	$L_{ij+1k1}$	$L_{ijk7}$
11	$L_{ijk7}$	$L_{ij+1k3}$	$L_{ijk+1 4}$
12	$L_{i+1jk5}$	$L_{ij+1k6}$	$L_{ijk+1 4}$

$P_a$  点に到達するまでの時間  $s$  は,

$$v(s) = b \tag{A.8}$$

となり、これを解いて、

$$s = \ln(b/a) / \beta \tag{A.9}$$

となる。

また、現像界面の位置  $x$  は、次式のようになる。

$$x(t) = a/\beta \cdot \{e^{\beta t} - 1\} \tag{A.10}$$

2. 稜線とセル面の関係

各ユニット  $U_{ijk}$  内に一意に 12 個のセル面  $\Sigma = \{F_{ijk\nu}, \nu=1\sim 12\}$  と 7 本の稜線  $E_g = \{L_{ijkm}, m=1\sim 7\}$  が定義できる。 $F_{ijk\nu}$  は 3 本の稜線から構成される。 $\Sigma$  と  $E_g$  の関係表 A.1 を示す。

(平成 2 年 3 月 28 日受付, 6 月 27 日再受付)



石塚 辰美

昭 50 北大・理・数学卒。昭 54 同大学院修士(数学)課程了。昭 57 同博士課程中退。同年(株)芙蓉情報センター(現・(株)富士総合研究所)入社。以来、技術計算用ソフトウェア開発に従事。昭 59 以後、半導体素子設計用シミュレータ開発に従事。現在、解析技術部

所属。