

修士論文

原子論的アプローチによるCu薄膜の  
初期形状が真性応力に与える影響の解析

p. 1 p. 70 完

平成20年 2月 8日 提出

指導教員 泉 聡志 准教授

66232 井藤 進也

# 目次

<b>1</b>	<b>序論</b>	<b>6</b>
1.1	研究の背景	6
1.1.1	真性応力問題	6
1.1.2	表面・界面応力の工学的応用	7
1.2	研究の目的	9
1.3	本論文の構成	10
<b>2</b>	<b>真性応力データベース</b>	<b>11</b>
2.1	緒言	11
2.2	真性応力データベース	12
2.3	考察	14
<b>3</b>	<b>基礎理論</b>	<b>15</b>
3.1	緒言	15
3.2	Kinetic Monte Carlo 法	15
<b>4</b>	<b>Kinetic Monte Carlo シミュレータの開発</b>	<b>17</b>
4.1	緒言	17
4.2	KMC パラメータ	18
4.2.1	遷移の可逆性の導入	18
4.2.2	EAM potential の導入	18
4.2.3	deposition のイベント	18
4.3	KMC パラメータの検証	19
<b>5</b>	<b>基板上での島の形状</b>	<b>22</b>
5.1	緒言	22
5.2	解析結果	23
5.3	結言	28

<b>6 真性応力の予測</b>	<b>29</b>
6.1 緒言	29
6.2 真性応力予測のためのパラメータ計算	30
6.3 真性応力予測	32
6.3.1 表面応力 $f$ の算出	32
6.3.2 形状の違いによる真性応力の予測	34
<b>7 結論と展望</b>	<b>37</b>
7.1 緒言	37
7.2 結論	37
7.3 展望	37
付録 A KMC プログラムソース	38
謝辞	67
参考文献	68

# 目 次

1.1	Illustration of the Volmer-Weber growth mode. (a) Before island coalescence. (b) After island coalescence . . . . .	7
2.1	The MOSS setup for measuring wafer curvature in real time . . . . .	14
4.1	Arrhenius plot of $\nu$ -to- $T$ for $M_i = 3$ . . . . .	20
4.2	Arrhenius plot of $\nu$ -to- $T$ for $M_i = 4$ . . . . .	20
5.1	initial state . . . . .	23
5.2	(a) final state . . . . .	24
5.3	(b) final state . . . . .	25
5.4	Existence rate of the number of atoms - Time plot for each coordination number $M_i$ of (a) . . . . .	26
5.5	Existence rate of the number of atoms - Time plot for each coordination number $M_i$ of (b) . . . . .	27
6.1	The initial state for KMC simulation . . . . .	30
6.2	The last state for KMC simulation (a) and (b) . . . . .	31
6.3	Schematic diagrams illustrate how to calculate surface stresses of singular surface . . . . .	32
6.4	Schematic diagrams illustrate how to calculate surface stresses with adatoms . . . . .	33
6.5	(a) last scene of deposition . . . . .	34
6.6	(b) last scene of deposition . . . . .	35

# 表 目 次

2.1	Database of intrinsic stress of SiN and AlN film . . . . .	12
2.2	Database of intrinsic stress of Si and other metal film . . . . .	13
4.1	Monte Carlo parameters . . . . .	19
6.1	Difference of $r$ and $h$ between (a) and (b) . . . . .	31

# 第 1 章 序論

## 1. 1 研究の背景

### 1. 1. 1 真性応力問題

現在の半導体デバイスは、多種多様な薄膜から構成されており、とりわけ近年の大規模集積回路の微細化に伴い、より高度な薄膜形成技術が要求されている。こうした中、形成された薄膜内に応力 (膜応力) が発生する問題が数十年も前より知られている<sup>1)</sup>。膜応力は、膜と基板の剥離・膜と基板界面からの転位の発生の駆動力となる。また、膜の電気的特性に著しい影響を及ぼし、膜質低下の大きな要因となる。そのため、膜応力制御は今尚深刻な課題となっている。よく知られた膜応力の発生要因の一つは熱応力であり、熱応力は、基板と薄膜材料の線膨張係数が既知であれば予測できる。しかしながら、観測される膜応力と熱応力とは一般に一致しない。なぜなら、熱応力とは別に、薄膜材料の基板上への堆積・成長という過程において、膜内に歪みが蓄積し応力が発生するためである。この応力は熱応力とは区別され、真性応力と称されている。

真性応力発生メカニズムの解明を試みる研究は、これまでも活発に行われており、Doerner ら<sup>2)</sup> や Koch<sup>3)</sup> は様々な膜種・膜製法・成長様式に応じた応力発生メカニズムを系統的に整理した。特に近年の真性応力問題の関心は、次世代半導体デバイスの薄膜材料として期待される多結晶薄膜、或いは大規模化が容易なアモルファス薄膜へと向けられている<sup>4)</sup>。今後、薄膜の厚みのナノオーダー化と薄膜の多層集積化がより一層進むことが予想され、薄膜形成過程の中でも成長初期に発生する真性応力メカニズムの解明が必要とされている。

多結晶薄膜やアモルファス薄膜は、基板上で Volmer-Weber 型成長 (以後 VW 成長) と呼ばれる三次元核成長をすることが知られている。Floro らのグループは、基板の反りを高精度で測定することで、VW 成長初期過程で生じる真性応力のリアルタイム (in-situ) 測定を実現した<sup>5)</sup>。彼らの実験から、成膜初期には圧縮応力が発生し、その後、膜が堆積するにつれて引張応力が生じるという傾向が得られている。Cammarata ら<sup>6)</sup> は、この初期の圧縮応力発生の有力なメカニズムの一つとしてキャピラリ応力 (Capillary-induced growth stress) を提案した。このメカニズムの基礎は、真性応力の発生要因を

表面・界面応力効果とする点にある．すなわち，VW 成長初期には，基板上に独立した微小な三次元核（結晶粒）が形成される．微小な核は表面応力と界面応力の影響を受けて歪み，表面積に依存した平均原子間距離を持つ．核がある程度大きくなると，界面との拘束が強くなって核は自由に変形できなくなるため，圧縮の真性応力が発生する．現在，キャピラリ応力の他にも，表面・界面応力効果に基づく圧縮応力の発生メカニズムがいくつか提案されており，活発な議論がなされている<sup>7-9)</sup>．一方，引張応力は，核同士の合体モデルで説明でき，そのメカニズムとして確立しつつある．初期核は，成長が進展するに伴い，核同士で合体を始める．合体の有無は表面エネルギーと界面エネルギーの釣り合いで決まり，合体時には核同士が弾性的変形を受ける．この時引張応力が発生する．この応力は，弾性論<sup>10),11)</sup> や有限要素法<sup>12),?)</sup> を用いた評価及び実験との比較が行われている．

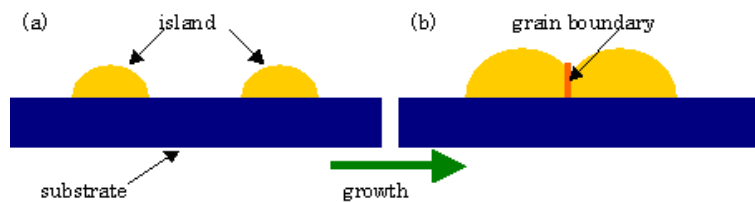


Fig. 1.1 Illustration of the Volmer-Weber growth mode. (a) Before island coalescence.  
(b) After island coalescence

このように，薄膜のマクロな機械的特性は，表面と界面の特性に強く支配される．表面と界面は基本的に不均質な部位であり，そこでの連続体近似の成立は疑わしい．よって，従来の連続体ベースの視点に止まらず，分子レベルでの視点から真性応力制御の問題に取り組むことが重要となる．本研究では，その第一歩として，真性応力予測のキーパラメータである，表面（界面）エネルギー・表面（界面）応力といった物理的特性を分子レベルから評価する．

### 1. 1. 2 表面・界面応力の工学的応用

表面（界面）応力は，固体表面のミクロな微視的構造とマクロな特性とを結ぶ固体表面特有の物理量である．歴史的には Gibbs<sup>14)</sup> によって熱力学的観点から定義され，その存在は古くから知られてきた．しかしながら，表面応力に強く依存した現象が注目されてきたのは，ナノテクノロジーの発展が著しい近年のこととあってよい．先に述べた真性応力も，表面応力効果のうちの一つであり，その制御と予測が大きな研究分野

として現在確立している．他，表面応力効果の例として，ナノ構造体の自己組織化<sup>15)</sup>，表面再構成<sup>16)</sup>，表面での化学反応<sup>17)</sup>といった現象が知られ，多くの研究がなされている．中でも近年，表面応力効果を逆にセンサーとして利用する研究がIBMを中心に精力的になされており，注目を集めている<sup>18)</sup>．基板表面に分子が吸着すると，表面応力が変化する．その時の応力変化を基板のそりとして計測することで，molという微量物質の検知が可能なセンサー開発が実現されている<sup>19)</sup>．同様の手法を用いて，Bashirら<sup>20)</sup>は1 nm/ pH の高精度なpH センサーを構築している．また，Fritzら<sup>21)</sup>は表面応力を利用して，DNA 生体分子量の検知に成功しており，表面応力はバイオテクノロジー分野への未曾有の貢献の可能性を秘めているといえる．



## 1. 2 研究の目的

前節で述べた様に、真性応力は、薄膜の微視的な成長様式、とりわけ、表面・界面の効果や成長初期の3次元島形状に強く依存するため、原子レベルからのプロセス理解が重要と考えられる。例えば、C. W. Paoら<sup>22)</sup>は、分子動力学法を用いて、基板上の2次元核が真性応力に及ぼす影響を計算している。しかしながら、より正確な真性応力予測のためには、3次元的な島形状の予測が重要と考えられる。

原子スケールでの膜成長のコンピュータシミュレーションとしては、分子動力学法(MD)や動的モンテカルロ法(KMC)などが挙げられる。分子動力学法は、直接的な数値解法であるものの、扱えるモデルの大きさや時間スケールに限られるため、秒スケールに及ぶ島の成長過程を直接予測することは困難である。一方、確率論的アルゴリズムを取り入れたKinetic Monte Carlo法は、秒単位・分単位といった、より大きなオーダの時間で進行する膜成長を扱う加速計算方法としては非常に有効である。そこで本研究では、このKinetic Monte Carlo法に着目する。

本研究では、Cu基板上Cu薄膜の初期成長過程における膜形状が、薄膜の真性応力に及ぼす影響を検討するため、島の結晶成長に適用可能なKinetic Monte Carloシミュレータを開発した。作成したKMCコードが、表面拡散の違いにより、様々な基板上で島の形状を再現することを示す。また、それに基づいて薄膜の真性応力を予測することを目的とする。

## 1. 3 本論文の構成

本論文は、本章を含め、全4章から構成される。以下に各章の概要を示す。

第2章 真性応力のデータベース では、他の研究グループによって行われている、真性応力測定実験に関する論文を収集、整理してデータベース化し、最近の真性応力研究の動向を探る。第3章 基礎理論 では、本研究で用いた手法の基礎的な原理を述べる。

第4章 Kinetic Monte Carlo シミュレータの開発 では、本研究で開発した Kinetic Monte Carlo シミュレータについて述べる。

第5章 基板上での島の形状 では、第4章で開発した KMC シミュレータを用いて、KMC パラメータの違いによる島の形状の再現をする。

第6章 真性応力の予測 では、表面・界面の特性値を、分子動力学法を用いて計算し、それらの値から真性応力の値を予測し、実験との比較を行う。

第7章 結論と展望 では、本研究のまとめをし、総括を述べる。

## 第 2 章 真性応力データベース

### 2. 1 緒言

本章では，研究の一環として集めた，真性応力に関する論文を整理してデータベース化し，最近の真性応力発生メカニズムの研究の動向について述べる．

## 2. 2 真性応力データベース

以下の Table 2.1 に、薄膜の材料を SiN, AlN とした真性応力研究の論文のデータベースを記す。本データベースは、筆者の卒業論文のために作成した真性応力データベースの一部を元に作成したものであり、表中の「論文 No.」は、そのデータベースにおける論文の番号である。

Table 2.1 Database of intrinsic stress of SiN and AlN film

論文No.	研究グループ	年	膜材料	基板材料	成膜方法	流入ガス	温度(圧力)	膜厚	測定方法	ウェハ直径	備考
1	108 M.Maeda, K.Ikeda(NTT System Electronics Lab.)	1998	SiN	Si or Quartz	rf-biased PCVD	SiH4-NH3-N2 mixture	400°C(360Pa) (測定: 20°C)	500nm	optically levered laser beam technique	6inch (.625 μ m thick)	Fig.1
2	202 Y.Toivola, J.Thurn, G.Cibuzar(Univ. of Minnesota)	2003	SiN	単結晶 Si(100), a-silica	LP(low-pressure)CVD	SiCl2H2(DCS), NH3	833°C(300 mTorr)	1 μ m	FSM900TC.Frontier Semiconductor Measurements, Inc., San Jose, CA	100mm (.500 μ m thick)	
3	203 M.Kammler(Univ. of Virginia), F.M.Ross(IBM)	2005	Si3N4	n-type Si(001)	LPCVD	DCS, NH3	780°C (6 × 10 <sup>-10</sup> dyn cm <sup>-2</sup> )	500nm		(100 μ m thick)	
4	24 S.Habermehl(Sandia National Lab.)	1998	Si3N4	n-type Si(100)	LPCVD	DCS, NH3	850°C(200 mTorr)	250~450nm	Tencor model FLX-2320	150mm	
5	195 E.Gianni(Istituto di Fotonica e Nanotecnologie IFN-CNR)	2005	SiN	low-doped single polished Si	ECR(electron cyclotron resonance)-PECVD	N2, SiH4, Ar	330°C	100~130nm	passive strain sensor	3inch	
6	18 B.W.Sheldon, A.Rajamani, E.Chason(Brown Univ.)	2005	AlN	Si(111)	ECR-MBE(molecular beam)	Al(, N2?)	750°C	400~1500nm	MOSS		
7	201 B.W.Sheldon, E.Chason(Brown Univ.)	2003	AlN	Si(111)	ECR-MBE(molecular beam)	Al(, N2?)	750°C	400~1500nm	MOSS		
8	宮永倫正(半導体技術研究所)	2006	AlN単結晶	SiC	昇華法		1900~2250°C(10~100kPa)	3 μ m ~4mm			
9	198 H.Edgar(Kansas State Univ.), M.Kuball(Univ. of Bristol, UK)	2002	AlN	Si polarity (0001) 6H-SiC	MO(metalorganic)CVD	Trimethylaluminum(TMA), NH3(H2)	1000°C(80Torr)	1.5 μ m	Raman scattering spectroscopy(Dilor XY micro-Raman system)		

以下の Table 2.1 に , 薄膜の材料を Si 及び他の金属元素とした真性応力研究の論文のデータベースを記す .

Table 2.2 Database of intrinsic stress of Si and other metal film

論文No.	研究グループ	年	膜材料	基板材料	成膜方法	流入ガス	温度(圧力)	膜厚	測定方法	ウェハ直径	備考
10	92 E.Chason, J.A.Floro(Sandia National Lab.)	1998	Si <sub>x</sub> Ge <sub>1-x</sub>	Si(001)	UHV deposition		760°C				Fig.2
11	56 B.W.Sheldon, A.Rajamani(Brown Univ.)	2001	polycrystalline diamond	Si(001)	CVD	hydrogen plasma, CH <sub>4</sub>	800°C(38 Torr)		laser-deflection		
12	105 J.A.Floro(Sandia National Lab.), E.Chason(Brown Univ.), S.C.Seel, C.V.Thompson(M.I.T.)	2001	a-Ge, p-Ge, Si, Ag, Al, Ti	Si(001)	electron beam evaporation		350, 750°C(10 <sup>-8</sup> Torr)	10~100nm	MOSS		Fig.3
13	111 J.A.Floro, S.C.Seel(Sandia National Lab.)	2003	a-Si, a-Ge	Si(001)	electron beam evaporation		25, 90, 150°C	40~nm	MOSS	(100 μ m thick)	
論文No.	研究グループ	年	膜材料	基板材料	成膜方法	流入ガス	温度(圧力)	膜厚	測定方法	ウェハ直径	備考
14	103 C.Friesen, S.C.Seel, C.V.Thompson(M.I.T.)	2004	polycrystalline Cu	amorphous substrates	e-beam evaporation				MOSS, capacitance, piezoresistive		
15	104 C.Friesen, C.V.Thompson(M.I.T.)	2002	polycrystalline Cu	borosilicate glass cantilever	e-beam evaporation		(3 × 10 <sup>-9</sup> Torr)		MOSS		
16	4 S.C.Seel, C.V.Thompson(M.I.T.), J.A.Floro(Sandia National Lab.)	2000	Ag	Si(001)	electron beam evaporation		30, 50, 100°C (10 <sup>-8</sup> Torr)			(100 μ m thick)	
17	115 S.J.Hearne, J.A.Floro(Sandia National Lab.)	2005	Ni	Si(100) capped with Ti/Au films	electrodeposition using a surfactant-free sulfamate		40~55°C		MOSS		Fig.4
18	186 M.Pletea(Leibniz-Institut für Festkörper und Werkstofforschung), R.Koch	2006	Co	oxidized Si(100)	magnetron-sputter deposition		(10 <sup>-6</sup> Pa)	300nm	laser-optical detection		

## 2.3 考察

前節で見たように，SiN 系薄膜では，主に DCS(Di Chloro Sirane) を流入ガスとした CVD(Chemical Vapor Deposition) による成膜技術が用いられている．また，AlN 系薄膜は，比較的高温での MBE(Molecular Beam Epitaxy) や CVD による研究が進んでいる．SiN を除く，AlN 系や Si 系，他金属の薄膜では，真性応力の測定方法として，MOSS(Multibeam Optical Stress Sensor) が多く用いられている．これは，次図に示すような，multibeam を用いて基板の反りを測定し，それから真性応力を算出するという方法<sup>23)</sup>である．リアルタイムに真性応力が測定できるため，現在主流の測定方法となっている．

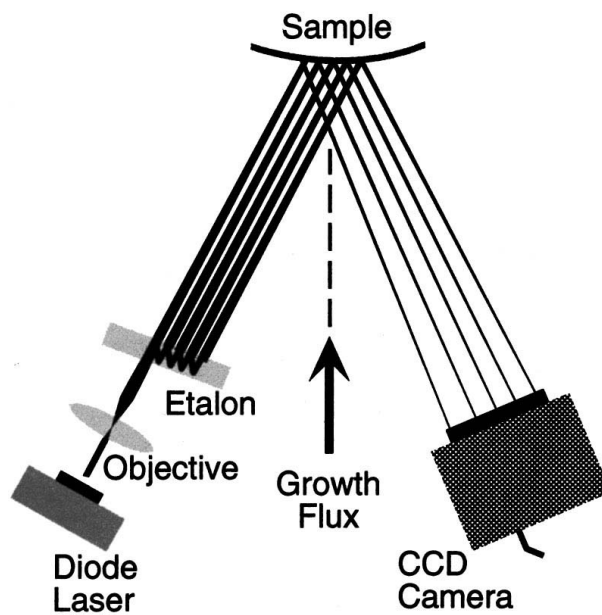


Fig. 2.1 The MOSS setup for measuring wafer curvature in real time

## 第 3 章 基礎理論

### 3. 1 緒言

本章では、本研究で用いた手法の基礎的な原理を述べる。

### 3. 2 Kinetic Monte Carlo 法

Kinetic Monte Carlo (KMC) 法は、乱数を用いて確率論的にイベントを発生させるため、個々の原子の運動方程式を数値的に解いて決定論的に時間発展させる分子動力学法に比べて、格段に早く膜成長のシミュレーションを進めることが出来る。

以下に、KMC の基本的なアルゴリズムを記す。

0) 初期配置として、対象とする系を格子で分割し、その格子上に原子を置く。

i) 各原子の配位数をリストアップする。

ii) 配位数が  $i$  の原子の遷移確率  $\nu_i$  を算出。

$E_a$  を活性化エネルギー、 $\nu_0$  を試行頻度 (prefactor) として、 $\nu_i$  は次のように表せる。

$$\nu_i = \nu_0 \exp\left(-\frac{E_a}{k_B T}\right) \quad (3.1)$$

ここで、 $k_B$  は Boltzmann、 $T[\text{K}]$  は基板温度である。

iii) 配位数が  $i$  の原子の個数  $N_i$  を求める。

iv) 全原子の遷移確率の和を  $R$  とし、

$$R = \sum_i N_i \nu_i \quad (3.2)$$

を算出する。

v) 配位数が  $i$  の原子のどれかが遷移する確率を  $p_i$  とすると、 $p_i$  は次式で表される。

$$p_i = \frac{N_i \nu_i}{R} \quad (3.3)$$

vi) 乱数  $x_{i_1}$  を発生させ,  $p_i$  の部分

$$P_j = \sum_i^j p_i \quad (3.4)$$

と比較し,  $P_{i-1} < \xi_1 < P_i$  となる配位数  $i$  を選択する.

vii) 乱数  $\xi_2$  を発生させ, 配位数が  $i$  の原子の中から 1 つの原子を選択する.

viii) 乱数  $\xi_3$  を発生させ, その原子の移動 (遷移) 先の空孔 (格子) を選択し, 移動させる.

ix) 乱数  $\xi_4$  を発生させ, 時間 (タイムステップ) を

$$\delta t = -\frac{\ln \xi_4}{R} \quad (3.5)$$

だけ進める.

x) i) に戻り, i) ~ ix) を繰り返す.



# 第 4 章 Kinetic Monte Carlo シミュレータ の開発

## 4. 1 緒言

本章では，前章で述べた Kinetic Monte Carlo (KMC) 法の原理に基づいて，本研究で開発した Kinetic Monte Carlo (KMC) シミュレータの概要を述べる．

## 4. 2 KMC パラメータ

KMC に関する文献

を拠り所として、前章で述べた基本の KMC のコードに、以下のような点を導入して、KMC シミュレータを開発した。

### 4. 2. 1 遷移の可逆性の導入

ある原子が配位数  $i$  の site にいて、隣接する site に遷移して配位数が  $j$  になったとすると、それぞれの状態の系のエネルギーを  $E_i$  及び  $E_j$  とすると、配位数が  $i$  から  $j$  に遷移する確率  $\nu_{i \rightarrow j}$  は、

$$\nu_{i \rightarrow j} = \nu_i \quad \text{if } E_j \leq E_i \quad (4.1a)$$

$$\nu_{i \rightarrow j} = \nu_i \exp\left(\frac{E_i - E_j}{k_B T}\right) \quad \text{if } E_j > E_i \quad (4.1b)$$

と表せる。

### 4. 2. 2 EAM potential の導入

前項で系のエネルギーが必要となるため、EAM potential を用いて、エネルギーを計算する。 $k$  ( $1 \leq k \leq N$ ) 番目の原子の配位数を  $M_k$  として、系全体のポテンシャルエネルギー  $E_{tot}$  は、

$$E_{tot} = \frac{1}{2} \sum_{k=1}^N \varphi M_k + \sum_{k=1}^N U(M_k) \quad (4.2)$$

で表される。これは、最近接原子との相互作用のみを考慮した簡略な式であるが、MD(分子動力学法) とのよい一致が得られている<sup>24)</sup>。

### 4. 2. 3 deposition のイベント

前章で述べた、全原子の遷移確率の和  $R$  を求めた上で、その値に deposition の頻度を加算することで、原子の遷移に加えて deposition のイベントの発生も考慮できるようにした。

### 4.3 KMCパラメータの検証

上述した各パラメータは，Gilmerらの論文<sup>24)</sup>を参照して，以下の値を用いた． $\varphi$ は相互作用の強さで $-0.15[\text{eV}]$ とし， $\nu_0$ ， $E_a$ ， $U$ については，下記の表で示す．

Table 4.1 Monte Carlo parameters

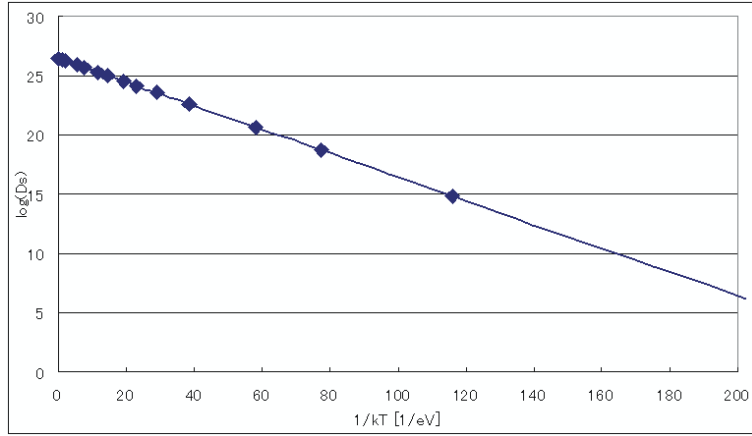
$M_i$	$\nu_0[\text{cm}^2/\text{s}]$	$E_a[\text{eV}]$	$U[\text{eV}]$
3	$2.0 \times 10^{-4}$	0.10	-1.960
4	$3.8 \times 10^{-1}$	0.35	-2.044
5	$4.0 \times 10^{-3}$	0.33	-2.153
6	$1.0 \times 10^{-2}$	0.60	-2.216
7	$1.0 \times 10^{-2}$	0.60	-2.248
8	$1.0 \times 10^{-2}$	0.60	-2.284
9	$1.0 \times 10^{-2}$	0.60	-2.303
10	$1.0 \times 10^{-2}$	0.60	-2.305
11	$1.0 \times 10^{-2}$	0.60	-2.378
12	—	—	-2.460

開発したコードを用いて，

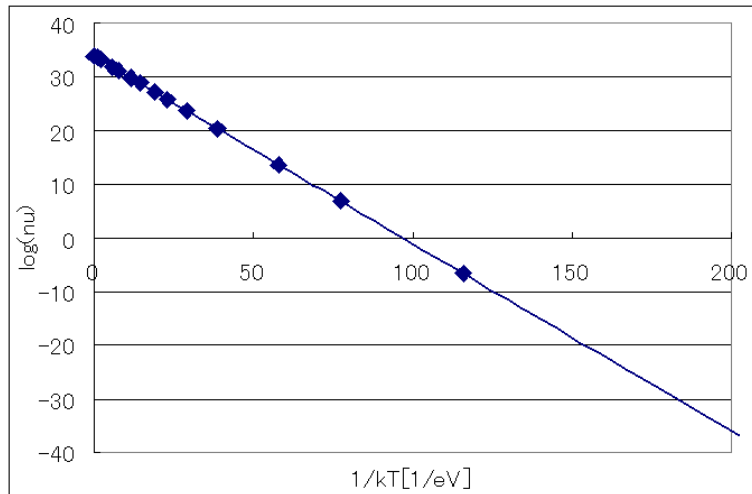
配位数が3の原子が，配位数が3の別の格子サイトに遷移する確率 $\nu_3$ ，

配位数が4の原子が，配位数が4の別の格子サイトに遷移する確率 $\nu_4$

の温度依存性を算出した (Fig.4.1,4.2) ．



**Fig. 4.1** Arrhenius plot of  $\nu$ -to- $T$  for  $M_i = 3$



**Fig. 4.2** Arrhenius plot of  $\nu$ -to- $T$  for  $M_i = 4$

$M_i = 3$  のとき, Fig.4.1 の直線の傾きから, 活性化エネルギー  $E_a$  は  $0.10[\text{eV}]$ , 縦軸の切片より,  $\nu_0$  は  $\exp(26.44) = 3.039 \times 10^{11}[\text{1/s}] = 1.99 \times 10^{-4}[\text{cm}^2/\text{s}]$  となり,

(1 回の遷移で  $2.556 \times 10^{-8}[\text{cm}]$  移動するため)

また,  $M_i = 4$  のとき, Fig.4.2 の直線の傾きから, 活性化エネルギー  $E_a$  は  $0.35[\text{eV}]$ , 縦軸の切片より,  $\nu_0$  は  $\exp(33.99) = 5.776 \times 10^{14}[\text{1/s}] = 3.77 \times 10^{-1}[\text{cm}^2/\text{s}]$  となり, Table 4.1 で記した設定パラメータと一致することが確認できた

## 第 5 章 基板上での島の形状

### 5. 1 緒言

本章では，前章で開発した KMC シミュレータを用いて，KMC パラメータの違いによる島の形状の再現をする．

## 5. 2 解析結果

表面拡散のパラメータが基板上的島形状に与える影響を検討した。

初期状態として 1300 個の原子を基板上に random に配置し，単位セル当たり  $40 \times 50$  原子の広さの基板上に存在する状態を初期条件とし (Fig.5.1)，基板温度を 600[K]，deposition なし (基板との相互作用にのみ注目するため) と設定して，KMC シミュレーションを実施した。

( 図中の色の分布は，基板に垂直方向の原子の層の高さを表す。 )

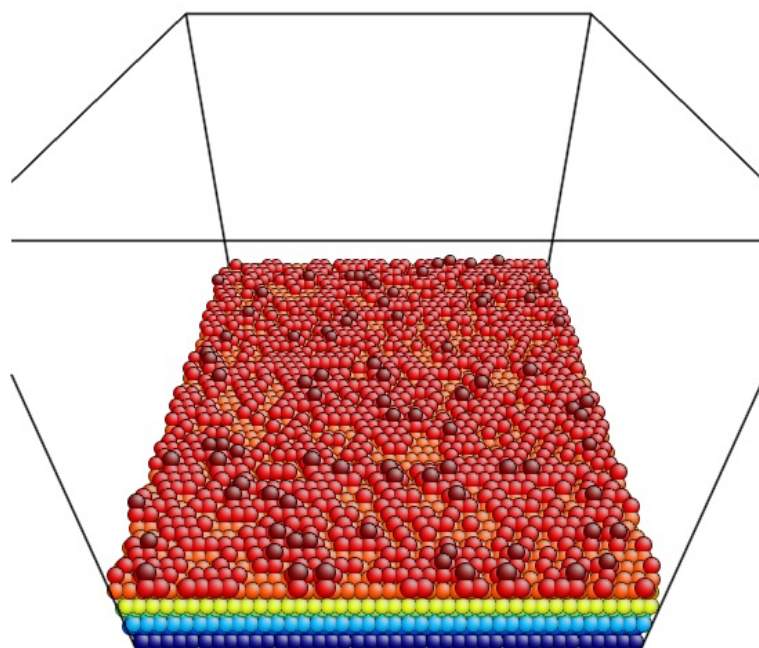


Fig. 5.1 initial state

Fig.5.2 に、基板上的 adatom が島の一層上に遷移する確率を 100 倍に、Fig.5.3 に、島にある adatom が一層下に遷移する確率を 100 倍とした時の島形状を示す。

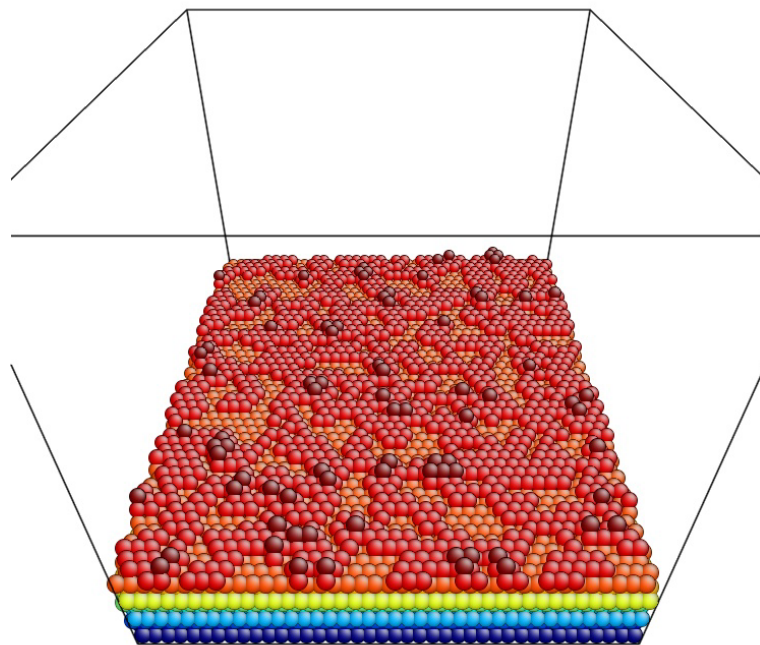


Fig. 5.2 (a) final state



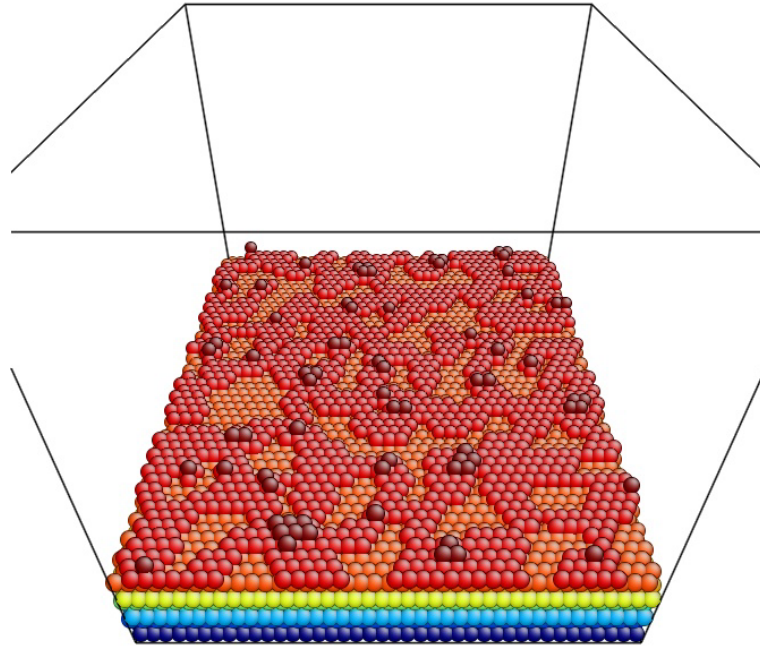
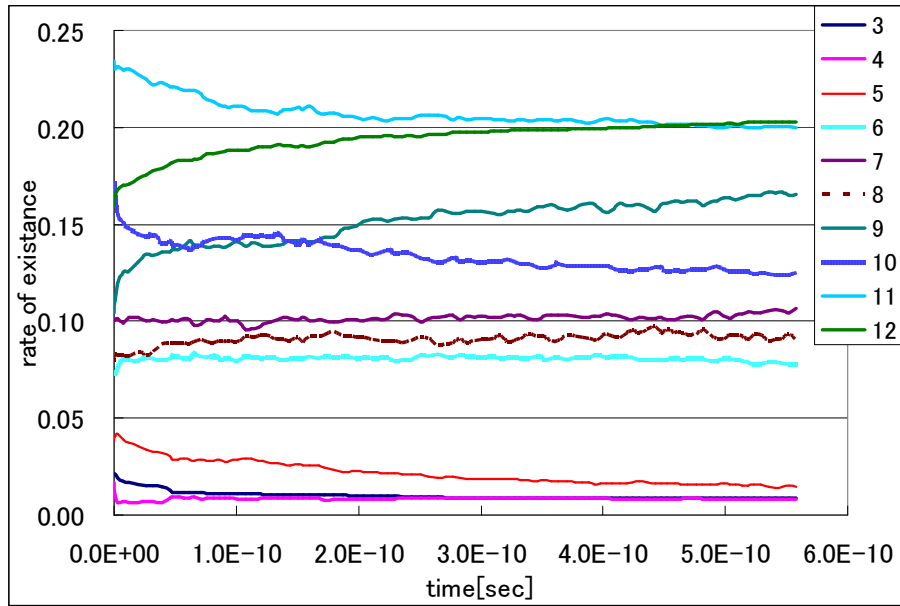
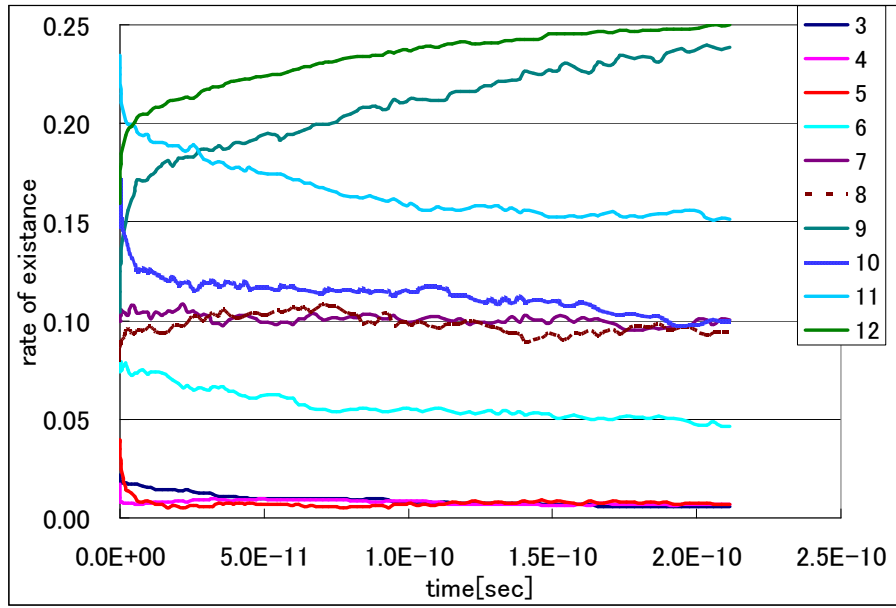


Fig. 5.3 (b) final state

表面拡散を表現するパラメータの違いにより，異なる様式の薄膜成長を再現できることが確認できた．また，これらの違いを評価するために，系内の adatom 及び基板表面の原子に対し，配位数ごと (3 ~ 12) の原子の存在比の時間経過による変化のグラフを Fig.5.4,5.5 に示す．



**Fig. 5.4** Existence rate of the number of atoms - Time plot for each coordination number  $M_i$  of (a)



**Fig. 5.5** Existence rate of the number of atoms - Time plot for each coordination number  $M_i$  of (b)

### 5. 3 結言

3 次元的な島を作りやすい (a) の成長様式では、配位数 6,10,12 の原子の存在比が高く、平坦な層状になりやすい (b) の成長様式では、凹凸の少なさを表現する配位数 9 の原子や 12 の原子の存在比が高くなっていて、基板上的 adatom の島の形状の特徴を示している。

表面拡散のパラメータの違いにより、表面原子の振る舞いは大きく異なり、その結果、島の形状が変化することがわかる。本研究で開発した KMC シミュレータにより、様々な成長初期段階の島形状が再現できるものと考えられる。

## 第 6 章 真性応力の予測

### 6. 1 緒言

前章までで述べたように，本研究で開発した KMC シミュレータにより異なる様式の薄膜成長を再現できることが確認できた．本章では，この KMC シミュレータで得られた島形状に対し真性応力を予測する．

## 6. 2 真性応力予測のためのパラメータ計算

表面拡散のパラメータの違いによる形状パラメータの変化を検討した．初期状態として 120 個の原子からなる島が，単位セル当たり  $40 \times 50$  原子の広さの基板の上に存在する状態を初期条件とし (Fig.6.1)，基板温度を 600[K]，deposition rate を  $5.0 \times 10^6$ [MonoLayer/sec] と設定し，KMC シミュレーションを実施した．

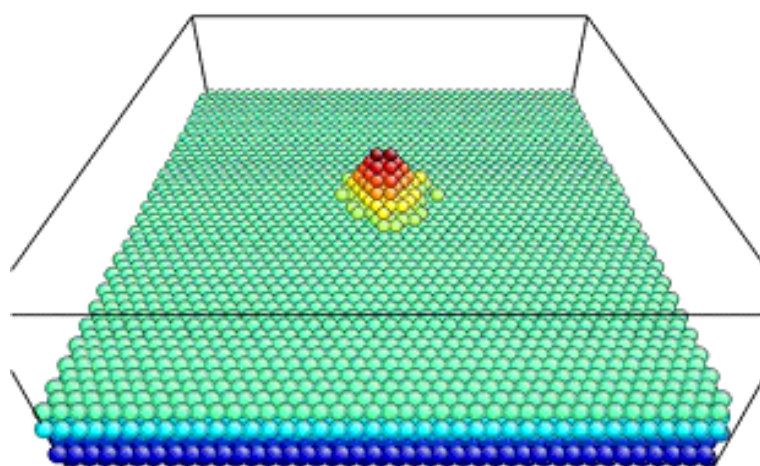
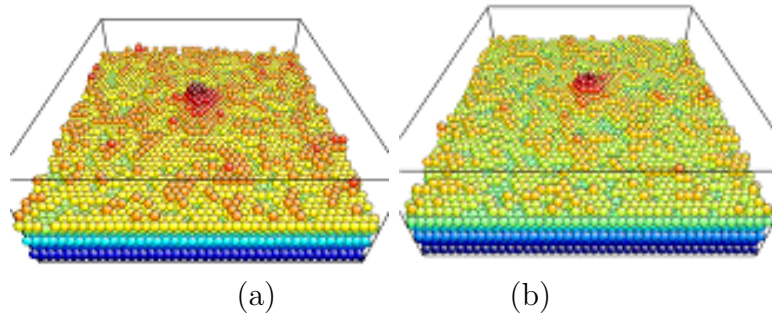


Fig. 6.1 The initial state for KMC simulation

Fig.6.2(a) に，基板上的 adatom が島の一層上に遷移する確率を 100 倍に，Fig.6.2(b) に，島にある adatom が一層下に遷移する確率を 100 倍とした時の島形状を示す．



**Fig. 6.2** The last state for KMC simulation (a) and (b)

これにより得られた得られた島の底面の半径  $r$  , 島の高さ  $h$  を Table 6.1 に示す . ここで得られる島のアスペクト比  $A$  は , 次節での真性応力を予測する際のパラメータとなる .

**Table 6.1** Difference of  $r$  and  $h$  between (a) and (b)

	$r[\text{\AA}]$	$h[\text{\AA}]$	$A = h/r$
(a)	14.8	10.2	0.689
(b)	13.5	7.65	0.567

## 6.3 真性応力予測

Cammarata ら<sup>25)</sup> によって、キャピラリ効果にともなう真性応力は、以下の式で与えられる。

$$\sigma = \frac{f+g}{A} \left( \frac{1}{r} - \frac{1}{r_{LD}} \right) \quad (6.1)$$

ここに、 $f$  は膜の表面応力、 $g$  は膜と基板との界面応力、 $A$  は、前節で計算した島の aspect 比 (島の高さ  $h$  と半径  $r$  の比)、 $r$  は島の半径、 $r_{LD}$  は島が基板から拘束を受け始める ("lockec-down" される) 半径である。

本研究の KMC シミュレータでは、Cu 基板上に単結晶状に Cu 原子が堆積するので、界面応力  $g$  は表面応力  $f$  に比べて十分小さく無視できると近似した。

### 6.3.1 表面応力 $f$ の算出

C. W. Pao ら<sup>22)</sup> によると、基板上的平坦な薄膜の表面応力  $f_0$  は、以下のようにして求められる。

Fig.6.3 のように、薄膜を面方向に歪ませた時のエネルギーの変化分  $\Delta U_0$  と (図中左)、半分の厚さの薄膜 2 枚を同様に面方向に歪ませた時のエネルギーの変化分  $\Delta U'_0$  を求める。

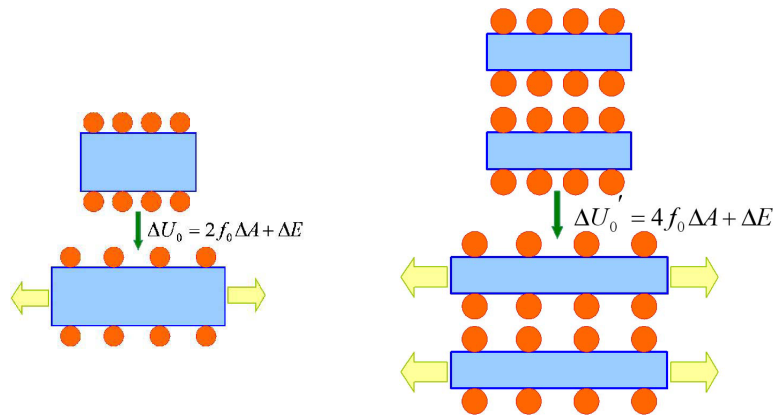


Fig. 6.3 Schematic diagrams illustrate how to calculate surface stresses of singular surface

これらより、凹凸のない薄膜の表面応力  $f_0$  は、 $\Delta A$  を歪による面積変化分として、次式で与えられる。

$$f_0 = \frac{1}{2} \frac{\Delta U'_0 - \Delta U_0}{\Delta A} \quad (6.2)$$



同様の手法を，片面に adatom が成長した薄膜に適用する．  
式 (6.2) で得た凹凸のない表面応力  $f_0$  の効果を打ち消すために，Fig.6.3,6.4 のように考えて，

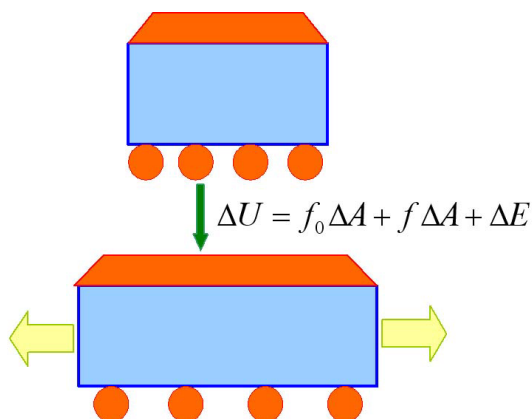


Fig. 6.4 Schematic diagrams illustrate how to calculate surface stresses with adatoms

$$f = f_0 + \frac{\Delta U - \Delta U_0}{\Delta A} \quad (6.3)$$

と求まる．

前章で算出した形状に対し，その原子情報を入力として分子動力学 (CG 法) シミュレーションを実施することにより，系のエネルギー及び表面応力を求めた．歪の値は  $\varepsilon = 0.01$  とした．

厚さ  $47.0[\text{\AA}]$ ，面方向 (周期境界)  $102 \times 110[\text{\AA}^2]$  の薄膜に対し，

$$\Delta U_0 = 2.81 \times 10^3 \text{ [eV]} \quad (6.4a)$$

$$\Delta U'_0 = 3.84 \times 10^3 \text{ [eV]} \quad (6.4b)$$

$$\Delta A = 227.5 \text{ [\AA}^2\text{]} \quad (6.4c)$$

$$f_0 = 1.41 \times 10^{-1} \text{ [J/m}^2\text{]} \quad (6.4d)$$

が得られた．

### 6. 3. 2 形状の違いによる真性応力の予測

deposition rate を  $1.0 \times 10^6$  [Monolayer/sec] , 基板温度を 600 [K] とし, 基板上に 8000 個 (4 [MonoLayer] 分) の原子が堆積するまで KMC シミュレーションを実施したところ,

(a) 基板上の adatom が島の一層上に遷移する確率を 100 倍,

(b) 島にある adatom が一層下に遷移する確率を 100 倍

とした時の島形状は Fig.6.5,6.6 のようになった .

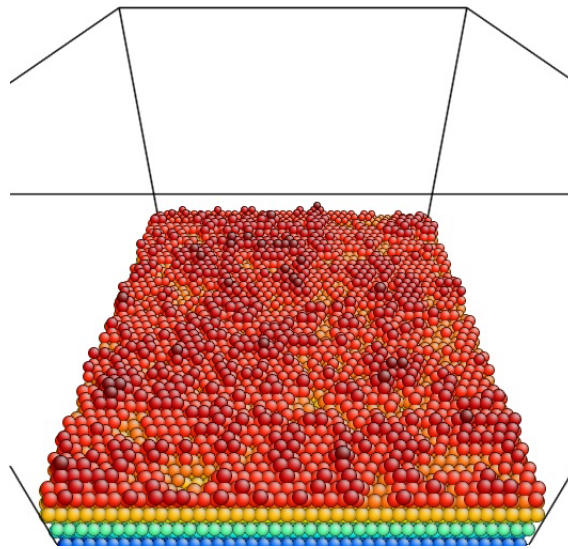
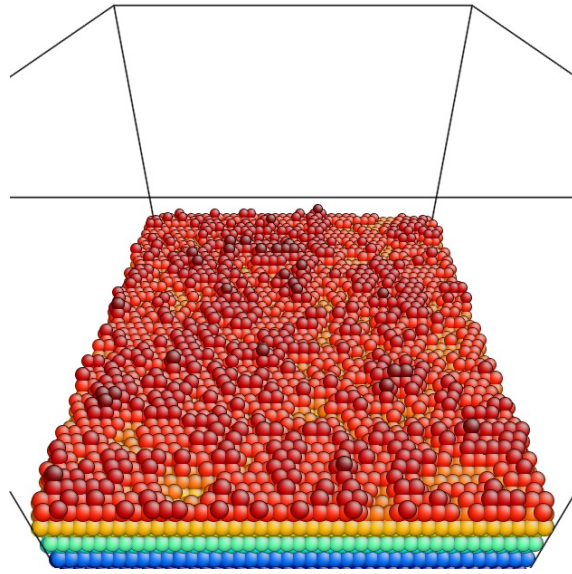


Fig. 6.5 (a) last scene of deposition



**Fig. 6.6** (b) last scene of deposition

これらの形状に対し，それぞれ， $r_{LD} = 5.0 \times 10^{-10}$  [m] と仮定した上で，  
 (a) :

$$\Delta U = 2.58 \times 10^3 [eV] \quad (6.5a)$$

$$f = -2.04 \times 10^{-1} [\text{J/m}^2] \quad (6.5b)$$

$$r = 1.50 \times 10^{-9} [\text{m}] \quad (6.5c)$$

$$\sigma = 362 [\text{MPa}] \quad (6.5d)$$

(b) :

$$\Delta U = 3.83 \times 10^3 [eV] \quad (6.6a)$$

$$f = -1.40 \times 10^{-1} [\text{J/m}^2] \quad (6.6b)$$

$$r = 1.00 \times 10^{-9} [\text{m}] \quad (6.6c)$$

$$\sigma = -280 [\text{MPa}] \quad (6.6d)$$

と求まった．

一般に，真性応力は，応力と膜厚の積で表される．膜厚を  $h = 4[\text{MonoLayer}] = 26.6[\text{\AA}]$  として，(a),(b) それぞれの真性応力  $\times$  膜厚は，

(a) :

$$\sigma \cdot h = 9.63 [\text{GPa} \cdot \text{\AA}] \quad (6.7)$$

(b) :

$$\sigma \cdot h = -7.45 [\text{GPa} \cdot \text{\AA}] \quad (6.8)$$

と算出される．これらは，J.A.Floro ら<sup>23)</sup> の実験値：10[GPa] と同じオーダーであることが確認できた．

## 第 7 章 結論と展望

### 7. 1 緒言

本章では、前章までで得られた結果をもとに、本研究の結論及び今後の展望を述べる。

### 7. 2 結論

KMC のシミュレータを開発し、島の結晶成長のモデルに適用した。その結果、表面拡散のパラメータの変化により、様々な島形状が再現できることがわかった。更に、本研究の KMC シミュレーションによって得られた基板上的堆積原子の島形状を MD (Conjugate Gradient 法) に受け渡して応力を計算することで、基板の表面応力を求めることができた。これに仮定した  $r_{LD}$  の値を与えることで、基板上的膜形状の違いによる真性応力の変化を予測することが可能となった。

### 7. 3 展望

仮定した部分の  $r_{LD}$  (島が基板から拘束を受け始める半径) に対しても、KMC シミュレーションを用いることで値を算出し、基板上的真性応力を予測することが可能になると考えられる。また、本研究で開発した KMC シミュレータを更に発展させれば、複数の島の合体のシミュレーションも出来るようになると考えられる。そして、従来使われている連続体ベースの真性応力の評価式の妥当性についても検討可能と考えられる。

## 付録A KMC プログラムソース

本研究で開発したKMCのプログラムのソースコードを以下に記す。

```
!*****
module mod_allocate_variables
implicit none
integer,public,allocatable :: latf(.,.,.)
integer,public,allocatable :: adatom(.,.)
integer,public,allocatable :: mask(,),mask2(,)
integer,public,allocatable :: ianum(,)
integer,public,allocatable :: ncd(,),ncdofnb(.,.)
integer,public,allocatable :: sos(.,.),ncd_ss(,)
real*8,public,allocatable :: x(,),y(,),z(,)
real*8,public,allocatable :: Energy(.,.)
real*8,public,allocatable :: Transitionrate(.,.)
real*8,public,allocatable :: dEne(.,.)
integer,public :: nnx,nnny,nnz,nss
integer,public :: nzfix,nztemp,nzfree
integer,public :: nfix,ntemp,nfree
integer,public :: iall,idep,isub,iad
integer,public :: iiend,iad_end
real*8,public :: Temp,kB
real*8,public :: rsum
real*8,public :: nu(12)
real*8,public :: nu_2,nu_3,nu_4,nu_5
real*8,public :: Eact(12)
real*8,public :: Fem(12),phi
real*8,public :: Deporate,DeporateML
real*8,public :: Strain

integer jj,kk

contains

subroutine allocate_variables
iiend=100000!20!2000 !60 ! 200
Temp=6.0d2 !Temperature[K]
kB=1.3806503d-23 !Boltzmann Constant 1.3806503d-23 [J/K]
kB=kB/1.6021773d-19 ! Boltzmann Const. 8.617337704d-5 [eV/K]

nnx=40!40!20 ! 20
nnny=50!50!24 ! even number 24
nzfix=3 ! layer 6
nztemp=3 ! layer 3
nzfree=3 ! layer 6
idep=35!10!8 ! layers of deposition (maximum)
nnz=nzfix+nztemp+nzfree+idep
! iad=10*12+8*11+6*10!120!30!1150!50!48 ! number of adatoms 100
iad=0

iad_end=8000!iad+10000

nfix=nnx*nnny*nzfix ! atoms
ntemp=nnx*nnny*nztemp ! atoms
nfree=nnx*nnny*nzfree ! atoms
```

```

isub=nfix+ntemp+nfree
iall=isub+iad
nss=nnx*nny ! number of atoms of substrate on surface
Deperate=2.0d9!1.0d10! ! 1/sec
DeperateML=Deperate/nss ! MonoLayer/sec
!   Deperate=DeperateML*nss
Strain=0.01d0
allocate(x(isub+iad+iiend+100000)
&       ,y(isub+iad+iiend+100000)
&       ,z(isub+iad+iiend+100000)
&       ,mask(isub+iad+iiend+100000)
&       ,mask2(isub+iad+iiend+100000)
&       ,ianum(isub+iad+iiend+100000))
allocate(latf(nnx,nny,nnz))
allocate(adatom(iad+iiend+100000,3))
allocate(ncd(iad+iiend+100000))
allocate(ncdofnb(iad+iiend+100000,12))
allocate(sos(nss+100000,3)) !lattice number of Subst On Surface
allocate(ncd_ss(nss+100000))
allocate(Energy(iad+iiend+100000,12))
allocate(dEne(iad+iiend+100000,12))
allocate(Transitionrate(iad+iiend+100000,12))
do jj=1,isub+iad+iiend
  x(jj)=0.0d0 ! [A]
  y(jj)=0.0d0 ! [A]
  z(jj)=0.0d0 ! [A]
  mask(jj)=0
  ianum(jj)=jj
enddo
do jj=1,iad+iiend
  do kk=1,3
    adatom(jj,kk)=0
  enddo
  ncd(jj)=0
  do kk=1,12
    ncdofnb(jj,kk)=0
    Energy(jj,kk)=0.0d0
    dEne(jj,kk)=0.0d0
    Transitionrate(jj,kk)=0.0d0
  enddo
enddo
do jj=1,nss
  do kk=1,3
    sos(jj,kk)=0
  enddo
enddo
do jj=1,2
  nu(jj)=1.0d-6 ! [cm^2/sec]
enddo
nu(3)=2.0d-4
nu(4)=3.8d-1
nu(5)=4.0d-3
do jj=6,12
  nu(jj)=1.0d-2
enddo
nu_2=1.0d0!5.0d0 ! tendency of adatom on substrate to move
nu_3=1.0d2!7.0d0 ! tendency of adatom on substrate to ascend
nu_4=1.0d2!3.0d0 ! tendency of adatom in island to ascend
nu_5=1.0d0      ! tendency of adatom in island to descend

```

```

do jj=1,2
  Eact(jj)=0.0d0 ! [eV]
enddo
Eact(3)=0.10d0
Eact(4)=0.35d0
Eact(5)=0.33d0
do jj=6,12
  Eact(jj)=0.6d0
enddo
!!
  Fem(0)=0.0d0
  Fem(1)=-1.0d0 ! embedding function [eV]
  Fem(2)=-1.5d0
  Fem(3)=-1.960d0
  Fem(4)=-2.044d0
  Fem(5)=-2.153d0
  Fem(6)=-2.216d0
  Fem(7)=-2.248d0
  Fem(8)=-2.284d0
  Fem(9)=-2.303d0
  Fem(10)=-2.305d0
  Fem(11)=-2.378d0
  Fem(12)=-2.460d0
  phi=-0.15d0 ! pair interaction strength [eV]
  return
end subroutine allocate_variables
end module mod_allocate_variables
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! start main program
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

program kmc_main
use mod_allocate_variables
implicit none
integer i,ii,ic
integer ix,iy,iz
integer ia
integer n_stgrnd
integer ixa,iya,iza

integer j,k
integer ianew,ixnew,iynew,iznew

integer intrand
integer CoordinationNumber ! (a)
integer iatemp ! (b)
integer ianx,iany,ianz ! (b),(c)
integer itr,itrmax ! (d)
integer mx,my,mz ! (a)
integer itest1,itest2
integer xmargin,ymargin,isl_x,isl_y

integer iss,mod2
integer icpx,icpy,icpz ! (b)
integer ipx,ipy,ipz,ipxx,ipyy,ipzz ! (b)
integer ncdip ! (b)
integer nsnap_cn,iisnap_cn
integer n_posi_Tr

integer island1
integer ini_switch

real*8 grnd,rand,rand2
real*8 cellx,celly,cellz

```



```

real*8 alatc
real*8 t,deltat,randt
real*8 rsum_ave ! except Deporate
real*8 center_island(3,3),width_island(3,3)
real*8 Ebef,Eaft(12) ! (b)
real*8 Etot      ! (b),(c)
real*8 trrnd     ! (d)

real*8 rt_cn(12)
integer*8 num_cn(12)

integer imv,idir ! (e)
integer imx,imy,imz ! (e)
!!!!!!
! initialize random number
call system_clock(count=ic)
call random_seed(put=(/ic/))
call random_number(rand2)
n_stgrnd = int(rand2*100000)
n_stgrnd = 22000
do i=1,n_stgrnd
    rand=grnd()
enddo
rand=rand*1.0d0
!!!!!!

alatc=3.614979d0 ! [A]
alatc=2.556162d0*sqrt(2.0d0) ! [A]
! i:number of atoms ii:time step
call allocate_variables
do jj=1,12
    nu(jj)=nu(jj)/(alatc*alatc*(1.0d-16)/2.0d0) ! [1/sec]
! write(6,'(i3,e14.6)')jj,nu(jj)
enddo

do j=1,iall
    ianum(j)=29
enddo

cellx=alatc*nnx/sqrt(2.0d0)
celly=alatc*nny*sqrt(6.0d0)/4.0d0
cellz=alatc*nnz*sqrt(3.0d0)/3.0d0
!!!!!!
! initialize lattice
do ix=1,nnx
    do iy=1,nny
        do iz=1,nnz
            latf(ix,iy,iz)=-1
        enddo
    enddo
enddo
!!!!!!
! initialize lattice next step
!!! write substrate
do ix=1,nnx
    do iy=1,nny
        do iz=1,nnz-idep
            latf(ix,iy,iz)=0
        enddo
    enddo
enddo
!!! write adatoms

```

```

center_island(1,1)=0.5d0!0.39d0!0.3d0 ! x center of 1st island
center_island(1,2)=0.5d0 ! y center of 1st island
width_island(1,1)=0.999999999d0!*0.01d0 ! x width of 1st island
width_island(1,2)=0.999999999d0!*0.01d0 ! y width of 1st island
center_island(2,1)=0.5d0!0.61d0!0.7d0 ! x center of 2nd island
center_island(2,2)=0.5d0 ! y center of 2nd island
width_island(2,1)=0.999999999d0!*0.01d0 ! x width of 2nd island
width_island(2,2)=0.999999999d0!*0.01d0 ! y width of 2nd island
island1=iad/2

if(iad.gt.0) then
do ia=1,iad
  ixa=0.999999999d0*grnd()*nnx+1
  iya=0.999999999d0*grnd()*nny+1
  if(ia.lt.island1) then
    ixa=( (center_island(1,1)-width_island(1,1)*0.5d0)
&          +(width_island(1,1)*grnd()) )*nnx+1
    iya=( (center_island(1,2)-width_island(1,2)*0.5d0)
&          +(width_island(1,2)*grnd()) )*nny+1
  endif ! ia<island1
  if(ia.ge.island1) then
    ixa=( (center_island(2,1)-width_island(2,1)*0.5d0)
&          +(width_island(2,1)*grnd()) )*nnx+1
    iya=( (center_island(2,2)-width_island(2,2)*0.5d0)
&          +(width_island(2,2)*grnd()) )*nny+1
  endif ! ia>=island1

  !!!! for NEB !!!!
  if(iiend==0) then
    if(ia==1) then
      ixa=19
      iya=26
    endif
    if(ia==2) then
      ixa=20
      iya=26
    endif
    if(ia==3) then
      ixa=21
      iya=25
    endif
  endif ! iiend==0
  !!!! for NEB !!!!

  ini_switch=1
  ini_switch=10
!
  if((ini_switch==1).or.(ini_switch==2))then
    if(ia==1) then
      ixa=nnx/2
      iya=nny/2+1
    endif
    if(ini_switch==2) then
      if(ia==2) then
        ixa=7
        iya=9
      endif
    endif
  endif ! switch==1,2
  if(ini_switch==5) then
    if(ia==1) ixa=5
    if(ia==2) ixa=3
    if(ia==3) ixa=4
    if(ia==4) ixa=7
    if(ia==5) ixa=8
    iya=9
  endif ! switch==5

```

```

if(ini_switch==10) then !! assuming nnz+idep+1=3k+1
  xmargin=15
  ymargin=17
  isl_x=10
  isl_y=12
  ixa=mod2(ia,isl_x)+xmargin
  iya=mod((ia-mod2(ia,isl_x))/isl_x,isl_y)+1+ymargin
  if((ia>isl_x*isl_y).and.
&      (ia<=isl_x*isl_y+(isl_x-2)*(isl_y-1))) then
    ixa=mod2(ia-isl_x*isl_y,isl_x-2)+xmargin+1
    iya=mod(((ia-isl_x*isl_y)-mod2(ia-isl_x*isl_y,isl_x-2))
&          /(isl_x-2),(isl_y-1))+2+ymargin
  endif
  if(ia>isl_x*isl_y+(isl_x-2)*(isl_y-1)) then
    ixa=mod2(ia-isl_x*isl_y-(isl_x-2)*(isl_y-1),isl_x-4)
&      +xmargin+2
    iya=mod(((ia-isl_x*isl_y-(isl_x-2)*(isl_y-1))
&          -mod2(ia-isl_x*isl_y-(isl_x-2)*(isl_y-1),isl_x-4))
&          /(isl_x-4),(isl_y-2))+2+ymargin
  endif
endif ! switch==10
iza=nnz-idep+1          !!!!!!!!!!!!!!!
iza=nnz-idep+3-mod(ia,3)      !!!!!!!!!!!!!!!
iza=nnz
!!!! for NEB !!!!
if(iiend==0) then
  if( (ia>=1).and.(ia<=3) ) then
    iza=nnz-idep+1
  endif ! 1<=ia<=3
endif ! iiend==0
!!!! for NEB !!!!

if((ini_switch>=1).and.(ini_switch<=9)) then
  iza=nnz-idep+1
endif ! 1<= ini_switc <=9
if(ini_switch==10) then
  iza=nnz-idep+1
  if((ia>isl_x*isl_y).and.
&      (ia<=isl_x*isl_y+(isl_x-2)*(isl_y-1))) then
    iza=nnz-idep+2
  endif
  if(ia>isl_x*isl_y+(isl_x-2)*(isl_y-1)) then
    iza=nnz-idep+3
  endif
!   write(6,'(4i6)')ia,ixa,iya,iza
  if(ia>isl_x*isl_y+(isl_x-2)*(isl_y-1)+(isl_x-4)*(isl_y-2))
&     write(6,'(a55)')
&     'ia>isl_x*isl_y+(isl_x-2)*(isl_y-1)+(isl_x-4)*(isl_y-2)'
endif

adatom(ia,1)=ixa
adatom(ia,2)=iya
adatom(ia,3)=iza
k=0
do while (latf(ixa,iya,iza) .gt. 0)
  k=k+1
  ixa=0.999999999999d0*grnd()*nnx+1
  iya=0.999999999999d0*grnd()*nny+1
  if(ia.lt.island1) then
&     ixa=( (center_island(1,1)-width_island(1,1)*0.5d0)
&         +(width_island(1,1)*grnd()) )*nnx+1
&     iya=( (center_island(1,2)-width_island(1,2)*0.5d0)

```

```

&          +(width_island(1,2)*grnd() )*nny+1
endif ! ia<island1
if(ia.ge.island1) then
  &      ixa=( center_island(2,1)-width_island(2,1)*0.5d0)
&          +(width_island(2,1)*grnd() )*nnx+1
&      iya=( center_island(2,2)-width_island(2,2)*0.5d0)
&          +(width_island(2,2)*grnd() )*nny+1
endif ! ia>=island1

  iza=idep
  iza=nnz-idep+1          !!!!!!!!!!!!!!!
  iza=nnz-idep+3-mod(ia,3)      !!!!!!!!!!!!!!!
  iza=nnz
  if(k.ge.10) iza=nnz
  adatom(ia,1)=ixa
  adatom(ia,2)=iya
  adatom(ia,3)=iza
enddo
latf(ixa,iya,iza)=ia
!      if (iza .ge. nnz-idep+2) then
&          do k=1,idep-1
&              call move0(iad,nnx,nny,nnz,latf,adatom,
&                  ixa,iya,iza,ia,iiend)
!          write(6,*)ia
&          enddo !k=1,idep-1
!          endif !if iza >= nnz-idep+2
!!         write(6,*)ia,adatom(ia,1),adatom(ia,2),adatom(ia,3)
enddo ! ia=1,iad
endif ! iad.gt.0

!      do iz=1,nnz
!          do ix=1,nnx
!              do iy=1,nny
!                  write(6,'(4i5)')ix,iy,iz,latf(ix,iy,iz)
!                  enddo
!              enddo
!          enddo
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

do iss=1,nss
  sos(iss,1)=mod2(iss,nnx)
  sos(iss,2)=(iss-sos(iss,1))/nnx+1
  sos(iss,3)=nnz-idep
enddo

!ccc open files
!      if(iii == 0) then
&          open(101,file='pv.dat')
&          open(102,file='mask.dat')
&          open(103,file='data.dat')
&          open(105,file='coord.dat')
&          open(106,file='energy1.dat')
&          open(107,file='Ds_kT.dat')
!
!ccc write header ccccccccccccccccccc
!      write(101,*)1,iall,koma ! atomic type, number of atoms, koma
&      write(101,*)1,iall,iiend+1
&      write(101,'(3f8.2)')cellx,cellz,celly ! lattice size
&      write(101,'(2f8.2)')0.0d0,1.0d0 ! initial time, time step
&      write(105,'(a38)')' step   time[s]   num_cn(k)   rt_cn(k)'
!
t=0.0d0
deltat=0.0d0
write(103,'(2e13.5)')t,deltat

```

```

      ii=0
!ccc  write initial state  cccc
      call write_coordinates(ii,iiend,iall,nnx,nyy,nnz,x,y,z,
&                                     mask,mask2,ianum,nfix,ntemp,nfree,
&                                     alatc,cellx,celly,cellz,latf,
&                                     nzfix,nztemp,nzfree,idep,iad,isub
&                                     ,Strain,iad_end)
!cccccccccccccccccccccccccccccccccccc
      if(iad==0) then
          write(6,'(a10)')'no adatom'
          stop
      endif ! iad==0

      Etot=0.0d0
      rsum_ave=0.0d0

      nsnap_cn=100 ! number of steps in "coord.dat"
      if(iiend<nsnap_cn)then
          nsnap_cn=iiend
      endif
      iisnap_cn=iiend/nsnap_cn

!!!!!!! start main loop !!!!!!!!
      step:do ii=1,iiend
          !!! start counting coordination number (a)
          do ia=1,iad
              ncd(ia)=CoordinationNumber(adatom(ia,1),adatom(ia,2)
&                                     ,adatom(ia,3))
              do j=1,12
                  ncdofnb(ia,j)
&                  =CoordinationNumber(
&                  mx(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
&                  ,my(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
&                  ,mz(adatom(ia,1),adatom(ia,2),adatom(ia,3),j) )
&                  -1
          !'-1' means decreasing by moving of original atom to neighbor
          if(ii==1) then
              if(latf(mx(adatom(ia,1),adatom(ia,2),adatom(ia,3),j),
&                  my(adatom(ia,1),adatom(ia,2),adatom(ia,3),j),
&                  mz(adatom(ia,1),adatom(ia,2),adatom(ia,3),j))==-1)then
!                  write(6,*)ia,j,ncd(ia),ncdofnb(ia,j)
              endif
          endif
          enddo ! j=1,12
          enddo ! 1<=ia<=iad
          do iss=1,nss
              ncd_ss(iss)=CoordinationNumber(sos(iss,1),sos(iss,2)
&                                     ,sos(iss,3))
&          enddo ! 1<=iss<=nss

          !!! end counting coordination number
          if(ii==1) then
              do jj=1,12
                  num_cn(jj)=0
                  rt_cn(jj)=0.0d0
              enddo
              do ia=1,iad
                  num_cn(ncd(ia))=num_cn(ncd(ia))+1
              enddo !ia=1,iad
              do iss=1,nss
                  num_cn(ncd_ss(iss))=num_cn(ncd_ss(iss))+1
              enddo
          endif
      enddo

```

```

        enddo
        do jj=1,12
            rt_cn(jj)=float(num_cn(jj))/float(iad+nss)
        enddo
        write(105,'(i5,e11.4,2i2,10i5,2e8.1,10e11.4,i5)')ii-1,t
&      , (num_cn(jj),jj=1,12), (rt_cn(jj),jj=1,12),iad+nss
    endif ! ii==1

!!! start calculating total energy (b)
    Etot=0.0d0
    do ia=1,iad
        Etot=Etot+Fem(ncd(ia))+0.5d0*phi*ncd(ia)
    enddo ! ia=1,iad
    do iss=1,nss
        Etot=Etot+Fem(ncd_ss(iss))+0.5d0*phi*ncd_ss(iss)
    enddo ! iss=1,nss
    if(ii==1) write(6,'(a6,e13.6)')'Etot= ',Etot
    do ia=1,iad
        icpx=adatom(ia,1) ! Center of Partial region
        icpy=adatom(ia,2)
        icpz=adatom(ia,3)
        Ebef=0.0d0
        if(adatom(ia,3)==nnz-idep+1) then
            do ipxx=icpx-3,icpx+3
                do ipyy=icpy-3,icpy+3
                    do ipzz=icpz-1,icpz+2
                        ipx=ipxx
                        ipy=ipyy
                        ipz=ipzz
                        !!! periodic boundary condition
                        if((ipxx<=0).or.(ipxx>=nnx+1)) then
                            ipx=mod2(ipxx,nnx)
                        endif ! ipxx
                        if((ipyy<=0).or.(ipyy>=nny+1)) then
                            ipy=mod2(ipyy,nny)
                        endif ! ipyy
                        if((ipzz>=1).and.(ipzz<=nnz)) then
                            if(latf(ipx,ipy,ipz)>=0) then
                                ncdip=CoordinationNumber(ipx,ipy,ipz)
                                Ebef=Ebef+Fem(ncdip)+0.5d0*phi*ncdip
                            endif ! latf>=0
                        endif ! 1<=ipzz<=nnz
                        !
                        !
                        !
                        if((ipzz<=0).or.(ipzz>=nnz+1)) then
                            ncdip=0
                        endif ! ipzz
                    enddo ! ipzz
                enddo ! ipyy
            enddo ! ipxx
        endif ! adatom(ia,3)==nnz-idep+1
        if(adatom(ia,3)>=nnz-idep+2) then
            do ipxx=icpx-3,icpx+3
                do ipyy=icpy-3,icpy+3
                    do ipzz=icpz-2,icpz+2
                        ipx=ipxx
                        ipy=ipyy
                        ipz=ipzz
                        !!! periodic boundary condition
                        if((ipxx<=0).or.(ipxx>=nnx+1)) then

```

```

        ipx=mod2(ipxx,nnx)
    endif ! ipxx
    if((ipyy<=0).or.(ipyy>=nny+1)) then
        ipy=mod2(ipyy,nny)
    endif ! ipyy
    if((ipzz>=1).and.(ipzz<=nnz)) then
        if(latf(ipx,ipy,ipz)>=0) then
            ncdip=CoordinationNumber(ipx,ipy,ipz)
            Ebef=Ebef+Fem(ncdip)+0.5d0*phi*ncdip
        endif ! latf>=0
    endif ! 1<=ipzz<=nnz
!
!
!
    if((ipzz<=0).or.(ipzz>=nnz+1)) then
        ncdip=0
    endif ! ipzz

    enddo ! ipzz
    enddo ! ipyy
    enddo ! ipxx
endif ! adatom(ia,3)>=nnz-idep+2

do j=1,12
    ianx=mx(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
    iany=my(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
    ianz=mz(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
    if( (latf(ianx,iany,ianz)==-1) .and.
&         (ncdofnb(ia,j).ge.3) ) then
!!!
!!! move atom temporarily
    latf(adatom(ia,1),adatom(ia,2),adatom(ia,3))=-1
    iatemp=ia
    latf(ianx,iany,ianz)=iatemp

    icpx=adatom(iatemp,1) ! Center of Partial region
    icpy=adatom(iatemp,2)
    icpz=adatom(iatemp,3)
    Eaft(j)=0.0d0
    if(adatom(iatemp,3)==nnz-idep+1) then
        do ipxx=icpx-3,icpx+3
        do ipyy=icpy-3,icpy+3
        do ipzz=icpz-1,icpz+2
            ipx=ipxx
            ipy=ipyy
            ipz=ipzz
            !!! periodic boundary condition
            if((ipxx<=0).or.(ipxx>=nnx+1)) then
                ipx=mod2(ipxx,nnx)
            endif ! ipxx
            if((ipyy<=0).or.(ipyy>=nny+1)) then
                ipy=mod2(ipyy,nny)
            endif ! ipyy
            if((ipzz>=1).and.(ipzz<=nnz)) then
                if(latf(ipx,ipy,ipz)>=0) then
                    ncdip=CoordinationNumber(ipx,ipy,ipz)
                    Eaft(j)=Eaft(j)+Fem(ncdip)+0.5d0*phi*ncdip
                    dEne(ia,j)=Ebef-Eaft(j)
                endif ! difference between Ebef & Eaft
            endif ! latf>=0
        endif ! 1<=ipzz<=nnz
        if((ipzz<=0).or.(ipzz>=nnz+1)) then
            ncdip=0
        endif ! ipzz
    endif ! adatom(iatemp,3)==nnz-idep+1
enddo

```

```

!               endif ! ipzz
                enddo ! ipzz
                enddo ! ipyy
                enddo ! ipxx
            endif ! adatom(ia,3)==nnz-idep+1
            if(adatom(iatemp,3)>=nnz-idep+2) then
                do ipxx=icpx-3,icpx+3
                do ipyy=icpy-3,icpy+3
                do ipzz=icpz-2,icpz+2
                    ipx=ipxx
                    ipy=ipyy
                    ipz=ipzz
                    !!! periodic boundary condition
                    if((ipxx<=0).or.(ipxx>=nnx+1)) then
                        ipx=mod2(ipxx,nnx)
                    endif ! ipxx
                    if((ipy<=0).or.(ipy>=nny+1)) then
                        ipy=mod2(ipy,nny)
                    endif ! ipy
                    if((ipzz>=1).and.(ipzz<=nnz)) then
                        if(latf(ipx,ipy,ipz)>=0) then
                            ncdip=CoordinationNumber(ipx,ipy,ipz)
                            Eaft(j)=Eaft(j)+Fem(ncdip)+0.5d0*phi*ncdip
                            dEne(ia,j)=Ebef-Eaft(j)
            !!!                ! difference between Ebef & Eaft
                            endif ! latf>=0
                            endif ! 1<=ipzz<=nnz
            !                if((ipzz<=0).or.(ipzz>=nnz+1)) then
            !                    ncdip=0
            !                endif ! ipzz

                enddo ! ipzz
                enddo ! ipyy
                enddo ! ipxx
            endif ! adatom(ia,3)>=nnz-idep+2
            !!!                !!! return temporary movement of atom
            latf(ianx,iany,ianz)=-1
            latf(adatom(ia,1),adatom(ia,2),adatom(ia,3))=ia
            endif ! latf==-1

            if( (latf(ianx,iany,ianz).ne.-1) .or.
            &                (ncdofnb(ia,j).le.2) ) then
                Eaft(j)=0.0d0
                dEne(ia,j)=Ebef-Eaft(j)
            endif ! latf.ne.-1

            if((ii==1).or.(ii==iiend)) then
            !                if(latf(mx(adatom(ia,1),adatom(ia,2),adatom(ia,3),j),
            !                &                my(adatom(ia,1),adatom(ia,2),adatom(ia,3),j),
            !                &                mz(adatom(ia,1),adatom(ia,2),adatom(ia,3),j))==-1)then
            !                    write(6,'(2i4,a5,e23.15)')ia,j,' dEne',dEne(ia,j)
            !                endif
            !                endif
                enddo ! j=1,12
            enddo !ia=1,iad
            !!! end calculating total energy
            !!! start calculating transition rate and sum (c)
            rsum=0.0d0
            n_posi_Tr=0

```



```

do ia=1,iad
  do j=1,12
    ianx=mx(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
    iany=my(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
    ianz=mz(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
    if( (latf(ianx,iany,ianz)==-1) .and.
&      (ncdofnb(ia,j).ge.3) ) then
      if(dEne(ia,j)>=0.0d0) then ! if Ebef>=Eaft
        Transitionrate(ia,j)=nu(ncd(ia))
&          *exp(-(Eact(ncd(ia)))/(kB*Temp))
      endif ! dEne>=0
      if(dEne(ia,j)<0.0d0) then ! if Ebef<Eaft
        Transitionrate(ia,j)=nu(ncdofnb(ia,j))!nu(ncd(ia))?
&          *exp(-(Eact(ncdofnb(ia,j))-dEne(ia,j))/(kB*Temp))
      endif ! dEne<0
!!!      !!! tendency of adatom on substrate to move
      if((j.ge.4).and.(j.le.9)) then
        if(adatom(ia,3)==nnz-idep+1) then
          Transitionrate(ia,j)=Transitionrate(ia,j)*nu_2
        endif ! adatom(ia,3)==nnz-idep+1
      endif ! 4<=j<=9
!!!      !!! tendency of adatom on sub , in isl to ascend
      if((j.ge.10).and.(j.le.12)) then
        if(adatom(ia,3)==nnz-idep+1) then
          Transitionrate(ia,j)=Transitionrate(ia,j)*nu_3
        endif ! adatom(ia,3)==nnz-idep+1
        if( (adatom(ia,3)>=nnz-idep+2).and.
&          (ncd(ia)<=6) ) then
          Transitionrate(ia,j)=Transitionrate(ia,j)*nu_4
        endif ! adatom(ia,3)>=nnz-idep+2
      endif ! 10<=j<=12
!!!      !!! tendency of adatom in isl to descend
      if((j.ge.1).and.(j.le.3)) then
&        if( (adatom(ia,3)>=nnz-idep+2).and.
          (ncd(ia)<=6) ) then
          Transitionrate(ia,j)=Transitionrate(ia,j)*nu_5
        endif
      endif ! 1<=j<=3
      if(ii==iiend)then
        if(latf(ianx,iany,ianz)==-1)then
!          write(6,*)ia,j,latf(ianx,iany,ianz)
        endif
      endif
    endif ! latf == -1
    if( (latf(ianx,iany,ianz).ne.-1) .or.
&      (ncdofnb(ia,j).le.2) ) then
      Transitionrate(ia,j)=0.0d0
      if(ii==iiend)then
!        if(latf(ianx,iany,ianz).ne.-1) then
!          write(6,*)latf(ianx,iany,ianz)
        endif
      endif
    endif !latf.ne.-1
    if(ii==iiend) then
!      if(ia==1) write(6,*)j,Transitionrate(ia,j)
    endif
!!!!!!! start eliminating Transition to emerge (C.N.<=2) atom
!!!!!!! and (C.N.>9) vacancy

```

```

ianx=mx(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
iany=my(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
ianz=mz(adatom(ia,1),adatom(ia,2),adatom(ia,3),j)
if( (latf(ianx,iany,ianz)==-1) .and.
&      (ncdofnb(ia,j).ge.3) ) then
!!!
!!! move atom temporarily
latf(adatom(ia,1),adatom(ia,2),adatom(ia,3))=-1
iatemp=ia
latf(ianx,iany,ianz)=iatemp
icpx=adatom(iatemp,1) ! Center of Partial region
icpy=adatom(iatemp,2)
icpz=adatom(iatemp,3)
if(adatom(iatemp,3)==nnz-idep+1) then
do ipxx=icpx-2,icpx+2
do ipyy=icpy-2,icpy+2
do ipzz=icpz-1,icpz+2
ipx=ipxx
ipy=ipyy
ipz=ipzz
!!! periodic boundary condition
if((ipxx<=0).or.(ipxx>=nnx+1)) then
ipx=mod2(ipxx,nnx)
endif ! ipxx
if((ipyy<=0).or.(ipyy>=nny+1)) then
ipy=mod2(ipyy,nny)
endif ! ipyy
if((ipzz>=1).and.(ipzz<=nnz)) then
if(latf(ipx,ipy,ipz)>=0) then
ncdip=CoordinationNumber(ipx,ipy,ipz)
if(ncdip<=2) then
Transitionrate(ia,j)=0.0d0
endif ! ncdip<=2
endif ! latf>=0
if(latf(ipx,ipy,ipz)==-1) then
ncdip=CoordinationNumber(ipx,ipy,ipz)
if(ncdip>=9) then
Transitionrate(ia,j)=0.0d0
endif
endif ! latf==-1
endif ! 1<=ipzz<=nnz
if((ipzz<=0).or.(ipzz>=nnz+1)) then
ncdip=0
endif ! ipzz

enddo ! ipzz
enddo ! ipyy
enddo ! ipxx
endif ! adatom(ia,3)==nnz-idep+1
if(adatom(iatemp,3)>=nnz-idep+2) then
do ipxx=icpx-2,icpx+2
do ipyy=icpy-2,icpy+2
do ipzz=icpz-2,icpz+2
ipx=ipxx
ipy=ipyy
ipz=ipzz
!!! periodic boundary condition
if((ipxx<=0).or.(ipxx>=nnx+1)) then
ipx=mod2(ipxx,nnx)
endif ! ipxx

```

```

        if((ipyy<=0).or.(ipyy>=nny+1)) then
            ipy=mod2(ipyy,nny)
        endif ! ipyy
        if((ipzz>=1).and.(ipzz<=nnz)) then
            if(latf(ipx,ipy,ipz)>=0) then
                ncdip=CoordinationNumber(ipx,ipy,ipz)
                if(ncdip<=2) then
                    Transitionrate(ia,j)=0.0d0
                endif ! ncdip<=2
            endif ! latf>=0
            if(latf(ipx,ipy,ipz)==-1) then
                ncdip=CoordinationNumber(ipx,ipy,ipz)
                if(ncdip>=9) then
                    Transitionrate(ia,j)=0.0d0
                endif
            endif ! latf==-1
        endif ! 1<=ipzz<=nnz
!
!
!
        if((ipzz<=0).or.(ipzz>=nnz+1)) then
            ncdip=0
        endif ! ipzz

        enddo ! ipzz
        enddo ! ipyy
        enddo ! ipxx
        endif ! adatom(ia,3)>=nnz-idep+2
!!!
        !!! return temporary movement of atom
        latf(ianx,iany,ianz)=-1
        latf(adatom(ia,1),adatom(ia,2),adatom(ia,3))=ia
        endif ! latf==-1
!!!!!!
        end eliminating Transition to emerge ...

        if(Transitionrate(ia,j).ne.0.0d0) then
!!!
            !!! if tr is positive value
            n_posi_Tr=n_posi_Tr+1
!
!
!
            if(ii==iiend) then
                write(106,'(2i4,e11.4)')ia,j,Transitionrate(ia,j)
            endif
            endif ! Tr>0.0d0
            rsum=rsum+Transitionrate(ia,j)

        enddo ! j=1,12
    enddo ! ia=1,iad

    if((ii==1).and.(iiend==1))then
!
!!
!!
!!
        write(6,'(a9,e14.6)')Tr(1,4)= ',Transitionrate(1,4)
        do jj=4,9
            write(6,'(e14.6)')Transitionrate(1,jj)
        enddo
        write(107,'(a6,a1,a11,a4,a8,a7,a6,a6)')',',',T',',',
&
        '1/kT',',',',Tr(1,5)',',',',ln(Tr)'
        write(107,'(4e14.6)')Temp,1.0d0/(kB*Temp)
&
        ,Transitionrate(1,5),log(Transitionrate(1,5))
    endif
    rsum_ave=rsum_ave+rsum
    rsum=rsum+Deperate

!!! end calculating transition rate and sum

!!! start selecting transition (d)
    itrmax=12*iad+1
    trrnd=rsum*grnd()*0.9999999999d0

!
!
!
    if(mod(ii,5)==0) then
        write(6,*)trrnd,rsum
    endif ! ii%5=0

```

```

    itr=1
    trrnd=trrnd-Transitionrate( itr/12+1-((12-mod(itr,12))/12
&      , mod(itr,12)+((12-mod(itr,12))/12)*12 )!/rsum
    do while((trrnd .ge. 0.0d0).and.(itr.le.itrmax-2))
        itr=itr+1
        trrnd=trrnd-Transitionrate( itr/12+1-((12-mod(itr,12))/12
&      , mod(itr,12)+((12-mod(itr,12))/12)*12 )!/rsum
    enddo ! while
    if(trrnd .ge. 0.0d0) then
        itr=itr+1
    endif ! trrnd>=0
!
!   &   write(6,*)itr,itr/12+1-((12-mod(itr,12))/12
!       ,mod(itr,12)+((12-mod(itr,12))/12)*12
    itest1=1
    itest2=1
    do k=1,60
        itest1=k/12+1-((12-mod(k,12))/12)
        itest2=mod(k,12)+((12-mod(k,12))/12)*12
        if(ii==iiend+1) then
            write(6,'(a4,i3,a5,i3)')'it1=',itest1,' it2=',itest2
        endif
    enddo !k=1,60

!!! end selecting transition
!!! start moving adatom (e)
if(itr.lt.itrmax) then
    imv=itr/12+1-((12-mod(itr,12))/12) !number of adatom to move
    idir=mod2(itr,12)
!
!   &   write(6,'(i3,a2,i4,a1,i3,a2,2i4)')ii,'(',imv,',',idir,
!       ')',ncd(imv),iad
    !imx means new x to which adatom of imv move
    imx=mx(adatom(imv,1),adatom(imv,2),adatom(imv,3),idir)
    imy=my(adatom(imv,1),adatom(imv,2),adatom(imv,3),idir)
    imz=mz(adatom(imv,1),adatom(imv,2),adatom(imv,3),idir)
    if(latf(imx,imy,imz).ne.-1) then
        write(6,*)ii,' , another atom already exists!'
!
!       write(6,*)imx,imy,imz,latf(imx,imy,imz)
!       stop
    endif ! latf.ne.-1
!
!       write(6,*)imx,imy,imz,latf(imx,imy,imz)
    latf(adatom(imv,1),adatom(imv,2),adatom(imv,3))=-1
    latf(imx,imy,imz)=imv
    adatom(imv,1)=imx
    adatom(imv,2)=imy
    adatom(imv,3)=imz
    if((ii.ge.iiend-1).and.(ii.le.iiend)) then
!
!       write(6,*)
!       do ia=1,iad
!           write(6,*)ia,adatom(ia,1),adatom(ia,2),adatom(ia,3)
!       enddo
!       write(6,*)
!       endif!(3<=ii<=4)
    endif ! itr<itrmax
    !!! end moving adatom

!!! start depositing new adatom (f)
if(itr.eq.itrmax) then
!
!       write(6,'(i3,a12,i4)')ii,' deposition',iad

```

```

!!!! new adatom start !!!!!
      iad=iad+1
      ianew=iad
      iall=iad+isub
      ianum(iall)=29

      ixnew=0.9999999999*grnd()*nnx+1
      iynew=0.9999999999*grnd()*nny+1

      iznew=nnz
      adatom(ianew,1)=ixnew
      adatom(ianew,2)=iynew
      adatom(ianew,3)=iznew
k=0
!   if(k<-1) then
do while (latf(ixnew,iynew,iznew) .gt. 0)
      k=k+1
      ixnew=0.9999999999*grnd()*nnx+1
      iynew=0.9999999999*grnd()*nny+1
      iznew=nnz
      if(k.ge.50) iznew=nnz
      adatom(ianew,1)=ixnew
      adatom(ianew,2)=iynew
      adatom(ianew,3)=iznew

!       write(6,*)'k=',k
enddo
!   endif!k<-1

      latf(ixnew,iynew,iznew)=ianew
!       write(6,*)ixnew,iynew
!       if (iznew .ge. nnz-idep+2) then
!           do k=1,idep-1
!               call move0(iad,nnx,nny,nnz,latf,adatom,
&                   ixnew,iynew,iznew,ianew,iiend)
!           enddo !k=1,idep-1
!       endif !if iznew >= nnz-idep+2
!!!!!! new adatom end !!!!!
endif ! itr==itrmax
!!! end depositing new adatom

if(ii==iiend) then
!   write(6,*)'Etot=',Etot
!   do ia=iad-5,iad
!       write(6,*)ia,ncd(ia)
!       do j=1,12
!           write(6,*)Energy(ia,j),Transitionrate(ia,j)
!       enddo
!   enddo
!   rsum_ave=rsum_ave/iiend
!   write(6,'(a10,e11.4,a13,e11.4,a11,e11.4)')
&       'rsum_ave = ',rsum_ave,' , Deporate = ',Deporate,
&       ' , r_ave = ',rsum_ave/float(n_posi_Tr)
endif!ii==iiend

      call write_coordinates(ii,iiend,iall,nnx,nny,nnz,x,y,z,
&                               mask,mask2,ianum,nfix,ntemp,nfree,
&                               alatc,cellx,celly,cellz,latf,
&                               nzfix,nztemp,nzfree,idep,iad,isub
&                               ,Strain,iad_end)

      randt=(1.0d0-1.0d-12)*grnd()+1.0d-12
      deltat=-log(randt)/rsum
      t=t+deltat

```

```

if(ii==1) write(103,'(2e13.5,i7)')t,deltat,iad
if(iiend>=20)then
if(mod(ii,20)==0)then
write(6,'(a45,i9)')
&
endif ! mod(ii,20)==0
endif ! iiend>=20

if(mod(ii,iisnap_cn)==0) then
do jj=1,12
num_cn(jj)=0
rt_cn(jj)=0.0d0
enddo
do ia=1,iad
num_cn(ncd(ia))=num_cn(ncd(ia))+1
enddo !ia=1,iad
do iss=1,nss
num_cn(ncd_ss(iss))=num_cn(ncd_ss(iss))+1
enddo
do jj=1,12
rt_cn(jj)=float(num_cn(jj))/float(iad+nss)
enddo
write(105,'(i5,e11.4,2i2,10i5,2e8.1,10e11.4,i5)')ii,t
&
,num_cn(jj),jj=1,12),(rt_cn(jj),jj=1,12),iad+nss
write(103,'(2e13.5)')t,deltat
endif ! mod(ii,iisnap_cn)==0
if(iad>=iad_end) exit
enddo step
!!!!!!! end main loop !!!!!!!!!!!!!

open(104,file='memo.dat')
write(104,'(a10,i5)')'iad',iad
write(104,'(a10,i5)')'iad_end',iad_end
write(104,'(a10,i5)')'iall',iall
write(104,'(a10,i5)')'nnx',nnx
write(104,'(a10,i5)')'nny',nny
write(104,'(a10,i5)')'nnz',nnz
write(104,'(a10,i5)')'iiend',iiend
write(104,'(a10,e12.5,a5)')'Temp',Temp,' [K]'
write(104,'(a10,e12.5,a7)')'rsum_ave',rsum_ave,' [1/s]'
write(104,'(a10,e12.5,a7)')'Deporate',Deporate,' [1/s]'
write(104,'(a10,e12.5,a8)')'DepoML',DeporateML,' [ML/s]'
write(104,'(a10,e12.5,a5)')'cellx',cellx,' [A]'
write(104,'(a10,e12.5,a5)')'celly',celly,' [A]'
write(104,'(a10,e12.5,a5)')'cellz',cellz,' [A]'
write(104,'(a10,e12.5,a5)')'alatc',alatc,' [A]'
write(104,'(a10,e12.5,a7)')'t',t,' [sec]'
write(104,'(a10,e12.5)')'thick_ave',
&
iall/(nnx*nny)*alatc/sqrt(3.0d0)
write(104,'(a10,e12.5)')'isl_dist',
&
(center_island(2,1)-center_island(1,1))*cellx
write(104,'(a10,e12.5)')'cent11',center_island(1,1)
write(104,'(a10,e12.5)')'cent21',center_island(2,1)
write(104,'(a10,e12.5)')'width11',width_island(1,1)
write(104,'(a10,e12.5)')'width21',width_island(2,1)
write(104,'(a10,e12.5)')'nu_2',nu_2
write(104,'(a10,e12.5)')'nu_3',nu_3
write(104,'(a10,e12.5)')'nu_4',nu_4
write(104,'(a10,e12.5)')'nu(2)',nu(2)

```

```

write(104,'(a10,e12.5)')nu(3)      ',nu(3)
write(104,'(a10,e12.5)')nu(4)      ',nu(4)
write(104,'(a10,e12.5)')nu(5)      ',nu(5)
write(104,'(a10,e12.5)')nu(6)      ',nu(6)
write(104,'(a10,e12.5)')nu(7)      ',nu(7)
write(104,'(a10,e12.5)')Strain      ',Strain
close(104)

close(101)
close(102)
close(103)
close(105)
close(106)
close(107)

if(ii==0) then
  write(6,*)intrand(1,10),CoordinationNumber()
endif !ii==0

!   Etot=0.0d0
!   iad=1000000
!   do ia=1,iad
!     t=0.999999999999d0*grnd()
!     Etot=Etot-log(t)
!   enddo
!   write(6,*)Etot/float(iad)
!   if(iad>=iad_end)write(6,'(a32)')'adatoms increased up to iad_end.'
!   write(6,'(a10)')'Completed.'

stop
end program kmc_main
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!   end main program
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!*****
integer function mx(cx,cy,cz,jdir)
use mod_allocate_variables
implicit none
integer,intent(in) :: cx,cy,cz,jdir

if((jdir==1).or.(jdir==4).or.(jdir==8).or.(jdir==11)) then
  if( ((mod(cy,2)==1).and.(mod(cz,3)==2))
& .or.((mod(cy,2)==0).and.(mod(cz,3)==1))
& .or.((mod(cy,2)==1).and.(mod(cz,3)==0)) ) then
    mx=cx-1
  endif

  if( ((mod(cy,2)==0).and.(mod(cz,3)==2))
& .or.((mod(cy,2)==1).and.(mod(cz,3)==1))
& .or.((mod(cy,2)==0).and.(mod(cz,3)==0)) ) then
    mx=cx
  endif
endif

if((jdir==2).or.(jdir==5).or.(jdir==9).or.(jdir==12)) then
  if( ((mod(cy,2)==1).and.(mod(cz,3)==2))
& .or.((mod(cy,2)==0).and.(mod(cz,3)==1))
& .or.((mod(cy,2)==1).and.(mod(cz,3)==0)) ) then
    mx=cx
  endif

  if( ((mod(cy,2)==0).and.(mod(cz,3)==2))
& .or.((mod(cy,2)==1).and.(mod(cz,3)==1))
& .or.((mod(cy,2)==0).and.(mod(cz,3)==0)) ) then
    mx=cx+1
  endif
endif

if(jdir==3) mx=cx

```

```

    if(jdir==6) mx=cx-1
    if(jdir==7) mx=cx+1
    if(jdir==10) mx=cx
    if(mx .ge. nnx+1) mx=mx-nnx
    if(mx .le. 0) mx=mx+nnx
end ! function mx
!*****
!*****
integer function my(cx,cy,cz,jdir)
use mod_allocate_variables
implicit none
integer,intent(in) :: cx,cy,cz,jdir
if((jdir==1).or.(jdir==2)) then
    if(mod(cz,3).ne.0) my=cy
    if(mod(cz,3).eq.0) my=cy+1
endif
if(jdir==3) then
    if(mod(cz,3).ne.0) my=cy-1
    if(mod(cz,3).eq.0) my=cy
endif
if((jdir==4).or.(jdir==5)) my=cy+1
if((jdir==6).or.(jdir==7)) my=cy
if((jdir==8).or.(jdir==9)) my=cy-1
if(jdir==10) then
    if(mod(cz,3).ne.2) my=cy+1
    if(mod(cz,3).eq.2) my=cy
endif
if((jdir==11).or.(jdir==12)) then
    if(mod(cz,3).ne.2) my=cy
    if(mod(cz,3).eq.2) my=cy-1
endif
if(my .ge. nny+1) my=my-nny
if(my .le. 0) my=my+nny
if(cx.lt.0) write(6,*)'x<0 ?'
end ! function my
!*****
!*****
integer function mz(cx,cy,cz,jdir)
use mod_allocate_variables
implicit none
integer,intent(in) :: cx,cy,cz,jdir
if((jdir.ge.1).and.(jdir.le.3)) then
    mz=cz-1
endif
if((jdir.ge.4).and.(jdir.le.9)) then
    mz=cz
endif
if((jdir.ge.10).and.(jdir.le.12)) then
    mz=cz+1
endif
if(mz .ge. nnz+1) mz=mz-nnz
if(mz .le. 0) mz=mz+nnz

```



```

        if(cx.lt.0) write(6,*)'x<0 ?'
        if(cy.lt.0) write(6,*)'y<0 ?'
    end ! function mz
!*****
!*****
    integer function CoordinationNumber(jx,jy,jz)
    use mod_allocate_variables
    implicit none
    integer,intent(in) :: jx,jy,jz
    integer mx,my,mz
    integer kdir
    integer ncn
    integer intrand

    ncn=0
    if((jz>=2).and.(jz<=nnz-1)) then
        do kdir=1,12
            if(latf(mx(jx,jy,jz,kdir),my(jx,jy,jz,kdir)
&                                     ,mz(jx,jy,jz,kdir)).ge.0) ncn=ncn+1
        enddo
    endif ! 2 <= jz <= nnz-1

    if(jz==1) then
        do kdir=4,12
            if(latf(mx(jx,jy,jz,kdir),my(jx,jy,jz,kdir)
&                                     ,mz(jx,jy,jz,kdir)).ge.0) ncn=ncn+1
        enddo
    endif ! jz == 1

    if(jz==nnz) then
        do kdir=1,9
            if(latf(mx(jx,jy,jz,kdir),my(jx,jy,jz,kdir)
&                                     ,mz(jx,jy,jz,kdir)).ge.0) ncn=ncn+1
        enddo
    endif ! jz == nnz

    if(iiend==0) then
        CoordinationNumber=intrand(1,12)
    endif

    CoordinationNumber=ncn

end
!*****
!*****
    subroutine move0(iad,nnx,ny,nnz,latf,adatom,
&                  ixa,iya,iza,ia,iiend)
    implicit none
    integer :: iad,iiend
    integer :: nnx,ny,nnz
    integer :: latf(nnx,ny,nnz)
    integer :: adatom(iad+iiend+100000,3)
    integer :: ixa,iya,iza,ia
    real*8 grnd,rand0

    integer :: jxalu,jyalu,jxaru,jyaru,jxad,jyad,jzam
! atoms that exist left_up side,right_up side,down side of ia atom
    integer :: ixa0,iya0,iza0
    integer :: k
! register 3 atoms on the under layer
    if(mod(iza,3)==2 .and. mod(iya,2)==0) then
        jxalu=ixa
        jyalu=iya
        jxaru=ixa+1
        jyaru=iya
        jxad=ixa

```

```

    jyad=iya-1
endif ! iza=2,iya=0
if(mod(iza,3)==2 .and. mod(iya,2)==1) then
    jxalu=ixa-1
    jyalu=iya
    jxaru=ixa
    jyaru=iya
    jxad=ixa
    jyad=iya-1
endif ! iza=2,iya=1
if(mod(iza,3)==1 .and. mod(iya,2)==1) then
    jxalu=ixa
    jyalu=iya
    jxaru=ixa+1
    jyaru=iya
    jxad=ixa
    jyad=iya-1
endif ! iza=1,iya=1
if(mod(iza,3)==1 .and. mod(iya,2)==0) then
    jxalu=ixa-1
    jyalu=iya
    jxaru=ixa
    jyaru=iya
    jxad=ixa
    jyad=iya-1
endif ! iza=1,iya=1
if(mod(iza,3)==0 .and. mod(iya,2)==0) then
    jxalu=ixa
    jyalu=iya+1
    jxaru=ixa+1
    jyaru=iya+1
    jxad=ixa
    jyad=iya
endif ! iza=0,iya=0
if(mod(iza,3)==0 .and. mod(iya,2)==1) then
    jxalu=ixa-1
    jyalu=iya+1
    jxaru=ixa
    jyaru=iya+1
    jxad=ixa
    jyad=iya
endif ! iza=0,iya=1
if(jxalu .ge. nnx+1) jxalu=jxalu-nnx
if(jxalu .le. 0) jxalu=jxalu+nnx
if(jyalu .ge. nny+1) jyalu=jyalu-nny
if(jyalu .le. 0) jyalu=jyalu+nny
if(jxaru .ge. nnx+1) jxaru=jxaru-nnx
if(jxaru .le. 0) jxaru=jxaru+nnx
if(jyaru .ge. nny+1) jyaru=jyaru-nny
if(jyaru .le. 0) jyaru=jyaru+nny
if(jxad .ge. nnx+1) jxad=jxad-nnx
if(jxad .le. 0) jxad=jxad+nnx
if(jyad .ge. nny+1) jyad=jyad-nny
if(jyad .le. 0) jyad=jyad+nny
jzam=iza-1
! move to the under layer
ixa0=ixa
iya0=iya
iza0=iza

```

```

k=0
if( (latf(jxalu,jyalu,jzam)==-1)
&    .and. (latf(jxaru,jyaru,jzam)==-1)
&    .and. (latf(jxad,jyad,jzam)==-1) .and. (k==0) ) then
rand0=grnd()*0.9999999999d0
iza=jzam
adatom(ia,3)=iza
if(rand0 .lt. 1.0d0/3.0d0) then
latf(ixa0,iya0,iza0)=-1
ixa=jxalu
iya=jyalu
adatom(ia,1)=ixa
adatom(ia,2)=iya
latf(ixa,iya,iza)=ia
endif ! if rand0 < 1/3
if((rand0 .ge. 1.0d0/3.0d0).and.(rand0 .lt. 2.0d0/3.0d0)) then
latf(ixa0,iya0,iza0)=-1
ixa=jxaru
iya=jyaru
adatom(ia,1)=ixa
adatom(ia,2)=iya
latf(ixa,iya,iza)=ia
endif ! if 1/3 < rand0 < 2/3
if(rand0 .ge. 2.0d0/3.0d0) then
latf(ixa0,iya0,iza0)=-1
ixa=jxad
iya=jyad
adatom(ia,1)=ixa
adatom(ia,2)=iya
latf(ixa,iya,iza)=ia
endif ! if 2/3 < rand0
k=1
endif ! if lu=-1,ru=-1,d=-1
if( (latf(jxalu,jyalu,jzam) >= 0)
&    .and. (latf(jxaru,jyaru,jzam)==-1)
&    .and. (latf(jxad,jyad,jzam)==-1) .and. (k==0) ) then
rand0=grnd()*0.9999999999d0
iza=jzam
adatom(ia,3)=iza
if(rand0 .lt. 1.0d0/2.0d0) then
latf(ixa0,iya0,iza0)=-1
ixa=jxaru
iya=jyaru
adatom(ia,1)=ixa
adatom(ia,2)=iya
latf(ixa,iya,iza)=ia
endif ! if rand0 < 1/2
if(rand0 .ge. 1.0d0/2.0d0) then
latf(ixa0,iya0,iza0)=-1
ixa=jxad
iya=jyad
adatom(ia,1)=ixa
adatom(ia,2)=iya
latf(ixa,iya,iza)=ia
endif ! if 1/2 < rand0
k=1
endif ! if lu>=0,ru=-1,d=-1
if( (latf(jxalu,jyalu,jzam)==-1)
&    .and. (latf(jxaru,jyaru,jzam) >= 0)

```

```

&      .and. (latf(jxad,jyad,jzam)==-1) .and.(k==0) ) then
rand0=grnd()*0.9999999999d0
iza=jzam
adatom(ia,3)=iza
if(rand0 .lt. 1.0d0/2.0d0) then
  latf(ixa0,iya0,iza0)=-1
  ixa=jxalu
  iya=jyalu
  adatom(ia,1)=ixa
  adatom(ia,2)=iya
  latf(ixa,iya,iza)=ia
endif ! if rand0 < 1/2
if(rand0 .ge. 1.0d0/2.0d0) then
  latf(ixa0,iya0,iza0)=-1
  ixa=jxad
  iya=jyad
  adatom(ia,1)=ixa
  adatom(ia,2)=iya
  latf(ixa,iya,iza)=ia
endif ! if 1/2 < rand0
k=1
endif ! if lu=-1,ru>=0,d=-1
if( (latf(jxalu,jyalu,jzam)==-1)
&      .and. (latf(jxaru,jyaru,jzam)==-1)
&      .and. (latf(jxad,jyad,jzam)>=0) .and.(k==0) ) then
rand0=grnd()*0.9999999999d0
iza=jzam
adatom(ia,3)=iza
if(rand0 .lt. 1.0d0/2.0d0) then
  latf(ixa0,iya0,iza0)=-1
  ixa=jxalu
  iya=jyalu
  adatom(ia,1)=ixa
  adatom(ia,2)=iya
  latf(ixa,iya,iza)=ia
endif ! if rand0 < 1/2
if(rand0 .ge. 1.0d0/2.0d0) then
  latf(ixa0,iya0,iza0)=-1
  ixa=jxaru
  iya=jyaru
  adatom(ia,1)=ixa
  adatom(ia,2)=iya
  latf(ixa,iya,iza)=ia
endif ! if 1/2 < rand0
k=1
endif ! if lu=-1,ru=-1,d>=0
if( (latf(jxalu,jyalu,jzam)==-1)
&      .and. (latf(jxaru,jyaru,jzam)>=0)
&      .and. (latf(jxad,jyad,jzam)>=0) .and.(k==0) ) then
latf(ixa0,iya0,iza0)=-1
ixa=jxalu
iya=jyalu
iza=jzam
adatom(ia,1)=ixa
adatom(ia,2)=iya
adatom(ia,3)=iza
latf(ixa,iya,iza)=ia
k=1
endif ! if lu=-1,ru>=0,d>=0

```

```

if( (latf(jxalu,jyalu,jzam)>=0)
& .and. (latf(jxaru,jyaru,jzam)==-1)
& .and. (latf(jxad,jyad,jzam)>=0) .and.(k==0) ) then
  latf(ixa0,iya0,iza0)=-1
  ixa=jxaru
  iya=jyaru
  iza=jzam
  adatom(ia,1)=ixa
  adatom(ia,2)=iya
  adatom(ia,3)=iza
  latf(ixa,iya,iza)=ia
  k=1
endif ! if lu>=0,ru=-1,d>=0
if( (latf(jxalu,jyalu,jzam)>=0)
& .and. (latf(jxaru,jyaru,jzam)>=0)
& .and. (latf(jxad,jyad,jzam)==-1) .and.(k==0) ) then
  latf(ixa0,iya0,iza0)=-1
  ixa=jxad
  iya=jyad
  iza=jzam
  adatom(ia,1)=ixa
  adatom(ia,2)=iya
  adatom(ia,3)=iza
  latf(ixa,iya,iza)=ia
  k=1
endif ! if lu>=0,ru>=0,d=-1
if( (latf(jxalu,jyalu,jzam)>=0)
& .and. (latf(jxaru,jyaru,jzam)>=0)
& .and. (latf(jxad,jyad,jzam)>=0) .and.(k==0) ) then
  k=1
endif
end subroutine move0
!*****
!*****
subroutine write_coordinates(ii,iiend,iall,nnx,nyy,nnz,x,y,z,
& mask,mask2,ianum,nfix,ntemp,nfree,
& alatc,cellx,celly,cellz,latf,
& nzfix,nztemp,nzfree,idep,iad,isub
& ,Strain,iad_end)
! use mod_allocate_variables
implicit none
integer,intent(in) :: ii
integer :: iiend,iad_end
integer :: iall
integer :: nnx,nyy,nnz
integer :: nzfix,nztemp,nzfree,idep
integer,intent(in) :: nfix,ntemp,nfree
integer,intent(in) :: iad,isub
real*8,intent(in) :: cellx,celly,cellz
integer :: latf(nnx,nyy,nnz)
real*8 :: x(isub+iad+iiend+100000),
& y(isub+iad+iiend+100000),z(isub+iad+iiend+100000)
integer :: mask(isub+iad+iiend+100000),
& mask2(isub+iad+iiend+100000)
integer :: ianum(isub+iad+iiend+100000)
integer :: koma,iisnap,nsnap
real*8 :: alatc
real*8,intent(in) :: Strain
real*8 route3,route2
integer :: i,ix,iy,iz

```

```

integer :: natomnum,nprognum
integer :: iclr
real*8 :: wmass
character(len=100) :: dirname
character(len=100) :: filename

nzfree=nzfree+0
route3=sqrt(3.0d0)
route2=sqrt(2.0d0)
i=0

do iz=1,nnz
  do iy=1,nnx
    do ix=1,nnx
      if (latf(ix,iy,iz).ge. 0) then
        i=i+1
        if(i .ge. 1) then
          if((mod(iz,3).eq.2).and.(mod(iy,2) .eq. 1)) then
            x(i)=(ix-1)*alatc/route2
            y(i)=(iy-1)*alatc*route2*route3/4.0d0
          endif
          if((mod(iz,3).eq.2).and.(mod(iy,2) .eq. 0)) then
            x(i)=(ix-1)*alatc/route2+alatc/(2*route2)
            y(i)=(iy-1)*alatc*route2*route3/4.0d0
          endif
          if((mod(iz,3).eq.1).and.(mod(iy,2) .eq. 1)) then
            x(i)=(ix-1)*alatc/route2+alatc/(2*route2)
            y(i)=(iy-1)*alatc*route2*route3/4.0d0
            & +alatc*route2*route3/12.0d0
          endif
          if((mod(iz,3).eq.1).and.(mod(iy,2) .eq. 0)) then
            x(i)=(ix-1)*alatc/route2
            y(i)=(iy-1)*alatc*route2*route3/4.0d0
            & +alatc*route2*route3/12.0d0
          endif
          if((mod(iz,3).eq.0).and.(mod(iy,2) .eq. 1)) then
            x(i)=(ix-1)*alatc/route2
            y(i)=(iy-1)*alatc*route2*route3/4.0d0
            & +alatc*route2*route3/6.0d0
          endif
          if((mod(iz,3).eq.0).and.(mod(iy,2) .eq. 0)) then
            x(i)=(ix-1)*alatc/route2+alatc/(2*route2)
            y(i)=(iy-1)*alatc*route2*route3/4.0d0
            & +alatc*route2*route3/6.0d0
          endif
          y(i)=celly-y(i)
          z(i)=(iz-1)*alatc*route3/3.0d0
          !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! idep !!!
          if((iz .ge.(nnz-idep+1)).and.(mod(iz,3)==0)) then
            mask(i)=i+300!14 !atomic color=red
          endif
          if((iz .ge.(nnz-idep+1)).and.(mod(iz,3)==1)) then
            mask(i)=i+600!13 !atomic color=red-red-orange
          endif
          if((iz .ge.(nnz-idep+1)).and.(mod(iz,3)==2)) then
            mask(i)=i+900!12 !atomic color=red-orange-orange
          endif
          ! if((iz .ge. 2).and.(mod(iz,3).eq.1))mask(i)=1
          ! if((iz .ge. 1).and.(iz .le. nzfix))mask(i)=1
          ! !atomic color=green
          if((iz .ge. 2).and.(mod(iz,3).eq.2))mask(i)=6
          if((iz .ge. (nzfix+nztemp)).and.(iz .le. (nnz-idep)))mask(i)=5
        endif
      endif
    enddo
  enddo
enddo

```

```

! atomic color=yellow
! if((iz .ge. 2).and.(mod(iz,3).eq.0))mask(i)=11
! if((iz .gt. nzfix).and.(iz .le. (nzfix+nztemp)))mask(i)=9
! atomic color=orange
! mask2(i)=latf(ix,iy,iz)
! mask2(i)=iz
!
! endif ! i .ge. 1
! endif ! latf(ix,iy,iz)>=0
! enddo ! ix=1,nnx
! enddo ! iy=1,nnx
! enddo ! iz=1,nnz
!
! write(6,'(i3,3f10.5)')ii,y(1),y(2),y(8)
! if( (ii==1).or.(ii==iiend) )write(6,'(f10.5)')y(1)
!! do i=1,iall
!! write(101,'(3f10.4)')x(i),cellz-z(i),y(i)
!! write(102,*)mask(i)
!! enddo
!
! iisnap=400
! nsnap=20 ! number of cfg files
! if(iiend<nsnap)then
! nsnap=iiend
! endif
! iisnap=iiend/nsnap
! if( (mod(ii,iisnap)==0)
& .or.((ii==0).and.(iiend==0)) ) then
!!! if(ii==0) then
! koma=ii/iisnap+1
! if((ii==0).and.(iiend==0)) koma=0
! write(6,'(a6,i5)')'koma =',koma
! write(dirname,'(a4)')'CFGs'
! write(filename,'(i5.5,".cfg")')koma
! write(6,*)dirname,filename
! open(201,file=trim(dirname)//trim(filename))
! open(201,file=trim(filename))
!-----
! output data for AtomEye format (extended CFG)
!-----
! write header information
! write(201,'(a22,i7)')'Number of particles = ',iall
! write(201,'(a4,f10.5,a30)')'A = ',1.0,
& ' Angstrom (basic length-scale)'
! write(201,'(a10,f10.5,a2)')'H0(1,1) = ',cellx,' A'
! write(201,'(a10,f10.5,a2)')'H0(1,2) = ',0.0,' A'
! write(201,'(a10,f10.5,a2)')'H0(1,3) = ',0.0,' A'
! write(201,'(a10,f10.5,a2)')'H0(2,1) = ',0.0,' A'
! write(201,'(a10,f10.5,a2)')'H0(2,2) = ',celly,' A'
! write(201,'(a10,f10.5,a2)')'H0(2,3) = ',0.0,' A'
! write(201,'(a10,f10.5,a2)')'H0(3,1) = ',0.0,' A'
! write(201,'(a10,f10.5,a2)')'H0(3,2) = ',0.0,' A'
! write(201,'(a10,f10.5,a2)')'H0(3,3) = ',cellz+5.0d0,' A'
! write(201,'(a13)')'.NO_VELOCITY.'
!
! write(201,'(a14,i3)')'entry_count = ',5
! write(201,'(a25)')'auxiliary[0] = color [ ] '
! write(201,'(a25)')'auxiliary[1] = color2 [ ] '
!-----
! natomnum=1
! nprognum=1
! natomnum=natomnum*1

```

```

        nprognum=nprognum*1
        wmass=63.55d0
        write(201,'(f10.5)')wmass
        write(201,'(a2)')'Cu)'
        do i=1,iall
            iclr=1
            if(i.gt.isub)iclr=(i-isub+1)*10
            iclr=iclr+0
!!        write(201,'(3e13.5,i4)')x(i)/cellx,y(i)/celly,z(i)/cellz,mask(i)
            write(201,'(3e13.5,2i6)')x(i)/cellx,y(i)/celly,
&            (cellz-z(i)-alatc*route3/3.0d0)/cellz,
&            i,!mask(i)
&            mask2(i) !colr2
            enddo
        close(201)
!-----
!      END output for AtomEye format
!-----
        endif    !! end of if (mod(ii,isnap)==0)

        if(( iad.ge.1).and.(ii==iiend) )
& .or.( iad==0).and.(ii==0) )
& .or.( ii==0).and.(iiend==0) )
& .or.( iad>=iad_end ) ) then
!-----
!      start writing 'Atom_posi.dat'
!-----
        open(301,file='Atom_posi.dat')
        open(302,file='Atom_lat.dat')
        open(303,file='st_Atom_posi.dat')
        open(304,file='st_Atom_lat.dat')

        write(301,*)iall
        write(301,*)nfree+iad,ntemp,nfix
        write(303,*)iall
        write(303,*)nfree+iad,ntemp,nfix
!!!!!! displacement
        do i=1,iall
            write(301,'(i2,6e21.13)')ianum(i),x(i),y(i),z(i)
&            ,z(i),0.0,0.0,0.0
            write(303,'(i2,6e21.13)')ianum(i)
&            ,x(i)*(1.0d0+Strain)
&            ,y(i)*(1.0d0+Strain)
&            ,z(i),0.0,0.0,0.0
            enddo

        write(301,*)iall
        write(301,*)
        write(303,*)iall
        write(303,*)
        do i=1,iall
            write(301,'(i2,3e21.13)')ianum(i),0.0,0.0,0.0
            write(303,'(i2,3e21.13)')ianum(i),0.0,0.0,0.0
        enddo

        write(302,'(3e21.13)')cellx,0.0,0.0
        write(302,'(3e21.13)')0.0,celly,0.0
        write(302,'(3e21.13)')0.0,0.0,cellz*2.5d0
        write(302,'(3e21.13)')cellx,0.0,0.0
        write(302,'(3e21.13)')0.0,celly,0.0
        write(302,'(3e21.13)')0.0,0.0,cellz*2.5d0
        write(304,'(3e21.13)')cellx*(1.0d0+Strain),0.0,0.0

```



```

        write(304,'(3e21.13)')0.0,celly*(1.0d0+Strain),0.0
        write(304,'(3e21.13)')0.0,0.0,cellz*2.5d0
        write(304,'(3e21.13)')cellx*(1.0d0+Strain),0.0,0.0
        write(304,'(3e21.13)')0.0,celly*(1.0d0+Strain),0.0
        write(304,'(3e21.13)')0.0,0.0,cellz*2.5d0

        close(304)
        close(303)
        close(302)
        close(301)
!-----
!      end writing 'Atom_posi.dat'
!-----
        endif    !! end of if (ii==iiend)

        end subroutine write_coordinates
!*****
!*****
        integer function mod2(a,b)
        implicit none
        integer,intent(in) :: a,b
        mod2=mod(a,b)+((b-mod(a,b))/b)*b
        end !function mod2
!*****
!*****
        integer function intrand(min,max)
        implicit none
        integer,intent(in) :: min,max
        real*8 grnd
        intrand=min+(max-min+1)*grnd()*0.9999999999d0
        end
!*****
!*****
        subroutine sgrnd(seed)
*
*      implicit integer(a-z)
*
*      Period parameters
        parameter(N      = 624)
*
*      dimension mt(0:N-1)
*              the array for the state vector
        common /block/mti,mt
        save   /block/
*
*      setting initial seeds to mt[N] using
*      the generator Line 25 of Table 1 in
*      [KNUTH 1981, The Art of Computer Programming
*      Vol. 2 (2nd Ed.), pp102]
*
        mt(0)= iand(seed,-1)
        do 1000 mti=1,N-1
            mt(mti) = iand(69069 * mt(mti-1),-1)
1000 continue
*
        return
        end
!*****
!*****
        double precision function grnd()
*
*      implicit integer(a-z)
*
*      Period parameters
        parameter(N      = 624)
        parameter(N1     = N+1)
        parameter(M      = 397)
        parameter(MATA   = -1727483681)

```

```

*           constant vector a
parameter(UMASK = -2147483648)
*           most significant w-r bits
parameter(LMASK =  2147483647)
*           least significant r bits
* Tempering parameters
parameter(TMASKB= -1658038656)
parameter(TMASKC= -272236544)
*
* dimension mt(0:N-1)
*           the array for the state vector
common /block/mti,mt
save  /block/
data  mti/N1/
*           mti==N+1 means mt[N] is not initialized
*
* dimension mag01(0:1)
* data mag01/0, MATA/
* save mag01
*           mag01(x) = x * MATA for x=0,1
*
TSHFTU(y)=ishft(y,-11)
TSHFTS(y)=ishft(y,7)
TSHFTT(y)=ishft(y,15)
TSHFTL(y)=ishft(y,-18)
*
* if(mti.ge.N) then
*           generate N words at one time
*   if(mti.eq.N+1) then
*           if sgrnd() has not been called,
*   call sgrnd(4357)
*           a default initial seed is used
*   endif
*   do 1000 kk=0,N-M-1
*       y=ior(iand(mt(kk),UMASK),iand(mt(kk+1),LMASK))
*       mt(kk)=ieor(ieor(mt(kk+M),ishft(y,-1)),mag01(iand(y,1)))
1000  continue
*   do 1100 kk=N-M,N-2
*       y=ior(iand(mt(kk),UMASK),iand(mt(kk+1),LMASK))
*       mt(kk)=ieor(ieor(mt(kk+(M-N)),ishft(y,-1)),mag01(iand(y,1)))
1100  continue
*   y=ior(iand(mt(N-1),UMASK),iand(mt(0),LMASK))
*   mt(N-1)=ieor(ieor(mt(M-1),ishft(y,-1)),mag01(iand(y,1)))
*   mti = 0
*   endif
*
*   y=mt(mti)
*   mti=mti+1
*   y=ieor(y,TSHFTU(y))
*   y=ieor(y,iand(TSHFTS(y),TMASKB))
*   y=ieor(y,iand(TSHFTT(y),TMASKC))
*   y=ieor(y,TSHFTL(y))
*
*   if(y.lt.0) then
*       grnd=(dble(y)+2.0d0**32)/(2.0d0**32-1.0d0)
*   else
*       grnd=dble(y)/(2.0d0**32-1.0d0)
*   endif
*
*   return
*   end

```

## 謝辞

本論文は、多くの方のご協力があって完成しました。

まず第一に、泉准教授，ありがとうございました。指導教員として以上に，最後の一年間は遠方から，様々な面でご指導いただき，学生として，人間として成長できたと思います。泉先生からの，瞬時で適切なアドバイスがあってこそ，研究が進捗したと思われます。

そして酒井教授，ありがとうございました。時に心配していただき，時に叱っていただき，研究以外の面でも強くなれました。

そして何より原助教，ありがとうございました。研究テーマを共有している人が研究室に少ない状況で，些細な質問にも親身に答えて下さり，日頃から研究の面倒を見ていただきました。こんな私が修士を修了できるのも，原先生がいらったからだと感謝しております。

研究室の皆様にも，色々助けていただいたり，ご迷惑を掛けてしまったり，そんな中で楽しく学生生活を送ることが出来ました。

私に関わった全ての人への感謝の意を込めて，本論文を終えたいと思います。

## 参考文献

- 1) 金原稔, 河野彰夫, 生地文也, 馬場茂, 薄膜の力学的特性評価技術 . Realize inc , 1992 .
- 2) M . F . Doemer and W . D . Nix . Stresses and deformation processes in thin films on substrates . CRC Critical Reviews in Solid and Materials Sciences , Vol . 14 , pp . 225-268 , 1988 .
- 3) R . Koch . The intrinsic stress of polycrystalline and epitaxial thin metal films . J . Phys . : Condens . Matter , Vol . 6 , pp . 9519-9550 , 1994 .
- 4) J . A . Floro , E . Chason , R . C . Cammarata , and D . J . Srolovitz . Physical origins of intrinsic stresses in volmer-weber thin films . MRS Bulletin , Vol . 27 , pp . 19-25 , 2002 .
- 5) J . A . Floro , S . J . Hearne , J . A . Hunter , P . Kotula , E . Chason , S . C . Seel , and C . V . Thompson . The dynamic competition between stress generation and relaxation mechanisms during coalescence of Volmer-Weber thin films . J . Appl . Phys . , Vol . 89 , pp . 4886-4897 , 2001 .
- 6) R . C . Cammarata and T . M . Trimble . Surface stress model for intrinsic stresses in thin films . J . Mater . Res . , Vol . 15 , pp . 2468-2474 , 2000 .
- 7) S . P . A . Gill , H . Gao , V . Ramaswamy , and W . D . Nix . Confined capillary stress during the initial growth of thin films on amorphous substrates . J . Appl . Mech . , Vol . 69 , pp . 425-432 , 2002 .
- 8) F . Spaepen . Interfaces and stresses in thin films . Acta mater . , Vol . 48 , pp . 31-42 , 2000 .
- 9) C . Friesen and C . V . Thompson . Reversible stress relaxation during precoalescence interruptions of Volmer-Weber thin film growth . Phys . Rev . Lett . , Vol . 89 , pp . 126103(1)-126103(4) , 2002 .

- 10) W . D . Nix and B . M . Clemens . Crystalline coalescence: A mechanism for intrinsic tensile stresses in thin films . *J . Mater . Res .* , Vol . 14 , pp . 3467-3473 , 1999 .
- 11) L . B . Freund and E . Chason . Model for stress generated upon contact of neighboring islands on the surface of a substrate . *J . Appl . Phys .* , Vol . 89 , pp . 4866-4873 , 2001 .
- 12) S . C . Seel , C . V . Thompson , S . J . Hearne , and J . A . Floro . Tensile stress evolution during deposition of Volmer-Weber thin films . *J . Appl . Phys .* , Vol . 88 , pp . 7079-7088 , 2000 .
- 13) S . C . Seel and C . V . Thompson . Tensile stress generation during island coalescence for variable island-substrate contact angle . *J . Appl . Phys .* , Vol . 93 , pp . 9038-9042 , 2003 .
- 14) J . W . Gibbs . The scientific papers of J . Willard Gibbs . London : Longmans Green , 1906 .
- 15) R . A . Kiehl , M . Yamaguchi , O . Ueda , N . Horiguchi , and N . Yokoyama . Patterned self-assembly of one-dimensional arsenic particle arrays in GaAs by controlled precipitation . *Appl . Phys . Lett .* , Vol . 68 , pp . 478-480 , 1996 .
- 16) R . J . Nichols , T . Nouar , C . A . Lucas , W . Haiss , and W . A . Hofer . Surface relaxation and surface stress of Au (111) . *Surf . Sci .* , Vol . 513 , pp . 263-271 , 2002 .
- 17) E . Kampshoff , E . Hahn , and K . Kern . Correlation between surface stresses and the vibrational shift of CO chemisorbed on Cu surfaces . *Phys . Rev . Lett .* , Vol . 73 , pp . 704-707 , 1994 .
- 18) A . Bietsch , J . Zhang , M . Hegner , H . Lang , and C . Gerber , Rapid functionalization of cantilever array sensors by inkjet printing . *Nanotechnology* , Vol . 15 , pp . 873-880 , 2004 .
- 19) R . Berger , E . Delamarche , H . P . Lang , C . Gerber , J . K . Gimzewski , E . Meyer , and H . Guntherodt . Surface stress in the self-assembly of alkanethiols on gold . *Science* , Vol . 276 , pp . 2021-2024 , 1997 .

- 20) R . Bashir , J . Z . Hilt , O . Elibol , A . Gupta , and N . A . Peppas . Micromechanical cantilever as ultrasensitive pH microsensor . Appl . Phys . Lett . , Vol . 81 , pp . 3091-3093 , 2002 .
- 21) J . Fritz , M . K . Baller , H . R . Lang , H . Rothuizen , P . Vettiger , E . Meyer , H . J . Guntherodt , C . Gerber , and J . K . Gimzewski . Translating biomolecular recognition into nanomechanics . Science , Vol . 288 , pp . 316-318 , 2000 .
- 22) C.W.Pao , D.J.Srolovitz , C.V.Thompson , "Effects of surface defects on surface stress of Cu(001) and Cu(111)" , Phys. Rev. Lett. , Vol.74 , 155437 (2006)
- 23) J.A.Floro , et al. , "The dynamic competition between stress generation and relaxation mechanisms during coalescence of Volmer-Weber thin films" , J.Appl.Phys. , Vol.89 , 4886 (2001)
- 24) H.Huang , G.H.Gilmer , et al. , "An atomistic simulator for thin film deposition in three dimensions" , J.Appl.Phys. , Vol.84 , 3636 (1998)
- 25) R.C.Cammarata , T.M.Trimble , "Surface stress model for intrinsic stresses in thin films" , J.Mater.Res. , Vol.15 , 2468 (2000)