

SiC 表面におけるエッジでの第一原理計算

情報科学科 西谷研究室 5613 安積 弘記

平成 21 年 2 月 20 日

概要

SiC は優れた物理的，化学的特徴を有することから Si に代わる次世代半導体材料として，近年注目されている．SiC は Si 半導体に比べ高耐圧，低損失でありさらに熱伝導率が高いことが知られている．このような優れた性質を持つ高品質 SiC の単結晶を作成する全く新しい原理に基づいた成長手法が，金子教授らによって発明された．これは液相成長法である準安定溶媒エピタキシー（Metastable Solvent Epitaxy）と呼ばれている．MSE は全く新しい成長法であり，その成長機構に関しては不明な点が多く面方位によって結晶成長の様子が異なる．MSE の SiC 単結晶成長において，その成長を律速する重要な機構が，表面とそれに直交する面が交差する「エッジ」である．

本研究は「エッジ」における原子レベルでの結晶成長の機構を理解することを目的とした．まず，3次元コンピュータグラフィックソフトウェアである Maya を用い，代表的な SiC 結晶である 3C, 4H, 6H-SiC の $\{0001\}$ ， $\{11-20\}$ ， $\{1-100\}$ 面が直交する格子モデルを作成し，さらに 4H-SiC の「エッジ」における結晶成長のアニメーションを作り C 原子が表面拡散をする経路の予想を立てた．次に，4H-SiC の $\{11-20\}$ ， $\{1-100\}$ 面での原子付着の第一原理計算を行った．結果， $\{11-20\}$ ， $\{1-100\}$ 面における最安定な場所が求まった．また， $\{11-20\}$ ， $\{1-100\}$ 面共にエネルギーバリアが大きく， $\{0001\}$ 面から C 原子が拡散することはほとんどなく， $\{11-20\}$ ， $\{1-100\}$ 面と $\{0001\}$ 面の「エッジ」が「つらら」の様な形で成長していくのではないかと結論に至った．

目次

第1章	序章	4
1.1	半導体炭化珪素の歴史	4
1.2	SiCの物性	5
1.3	MSE	7
1.3.1	構成	7
1.3.2	駆動力	7
1.3.3	成長機構	8
第2章	SiC表面の視覚化	9
2.1	SiCの構造を作成	9
2.1.1	Mayaについて	9
2.1.2	Rubyについて	9
2.1.3	RubyからMELへの変換	9
2.1.4	具体的なRubyスクリプト	10
2.1.5	SiCの3DCGモデル	14
2.1.6	SiCエッジの3DCGアニメーション	17
第3章	SiC表面での原子付着の第一原理計算	22
3.1	計算原理	22
3.2	計算手法	22
3.2.1	MedeAについて	23
3.2.2	C原子を付着させる手順	23
3.2.3	非平面派基底擬ポテンシャル法	23
3.2.4	C原子の付着位置	23
3.3	計算結果の考察	26
3.3.1	計算結果(最安定エネルギー)	26
3.3.2	計算結果(最安定位置)	27
3.3.3	考察	30

第1章 序章

1.1 半導体炭化珪素の歴史

半導体炭化珪素 (SiC) は, Si と C という地球上に多く存在する元素から構成されているにもかかわらず, 天然には隕石中にわずかに見出せるだけで, 大きな結晶塊は存在しない. しかし, 研究上では 19 世紀の初めにはその存在が知られており, 歴史的には最も古い化合物半導体の一つである. SiC が半導体として研究されるようになったのは, 1955 年に昇華法 (Lery 法) により高純度の SiC 単結晶が得られるようになってからである. しかし, レーリー法は自然核発生を利用する結晶成長技術であるために, 結晶の大きさは 10mm 角ほどで, 形も不揃いであったため, 工業的には適さなかった. また, Si 素子の驚異的な発展により, SiC 素子の研究開発は 1970 年代には下火になってしまった. しかし, 1981 年に Tairov らによって開発された改良レーリー法により大きな 6H-SiC 単結晶のインゴットが得られるようになる. 改良レーリー法は, 温度勾配を設けた成長系内を不活性ガスで満たすことにより原料の輸送過程を制御し, さらに種結晶を使うことにより結晶成長の各生成過程を制御することがバルク結晶成長法として大きなブレイクスルーとなった. 一方, エピタキシャル成長においては, 1987 年に $\{0001\}$ Si 面から数度傾いた結晶面を用いることによりステップでの成長を促進させる「ステップ制御エピタキシー」が開発され, 成長ポリタイプの制御, 成長表面の平坦化と成長表面の低温化が達成された. これと同時期に米国のベンチャー企業から 6H-SiC ウエハの市販が始まり, 良質な基盤結晶とエピタキシャル膜が得られるようになったことで, SiC は半導体素子用材料として再び注目されるようになった. 素子化技術においては, 1980 年代以降, SiC は多くの青色発光候補材料で困難であった pn 接合が可能であったことから, 青色発光ダイオード素子の有力な材料として研究開発が進められていた. しかし, 1993 年に窒素化合物半導体で高輝度青色発光素子が開発されると, 間接遷移型半導体である SiC は, 研究対象からはずれ, その

後は、高パワー、高周波パワーデバイスが主な研究開発のターゲットとなった。

1.2 SiC の物性

SiC は多くの結晶多形が存在し、それぞれ物性が異なる。SiC は Si と比べて約 3 倍バンドギャップが大きく、約 4 倍の熱伝導度と熱の放射能力が大きい。そのため、大電力制御の半導体デバイスの適用において、高温まで安定な動作が可能で、破壊に対する耐量大きい。また、絶縁破壊電解が Si に比べると大きく、Si パワーデバイスでは困難だった大容量と高速動作の両立が可能となる。3C-SiC はポリタイプの中で唯一、立方晶をとり、結晶の対象性が高く、等方的であるため、デバイス設計が容易である。また、低温において良質な結晶が作製可能なため、大面積化によるコスト低減や、さまざまなデバイスへの応用などが魅力的である。また、従来デバイス化研究 6H-SiC が用いられてきたが、4H-SiC の方が移動度は高く、移動度の異方性が小さいことから、パワーデバイス作製時に高い性能が期待される。また、SiC は極性結晶であり、デバイス作製時に {0001} 面を使う場合には {0001}Si 面と {000-1}C 面の選択がある。面方位によって SiC の結晶成長や酸化速度などが異なることや、成長の様相が違う。そのためデバイス設計には異方性も考慮しなくてはならない。

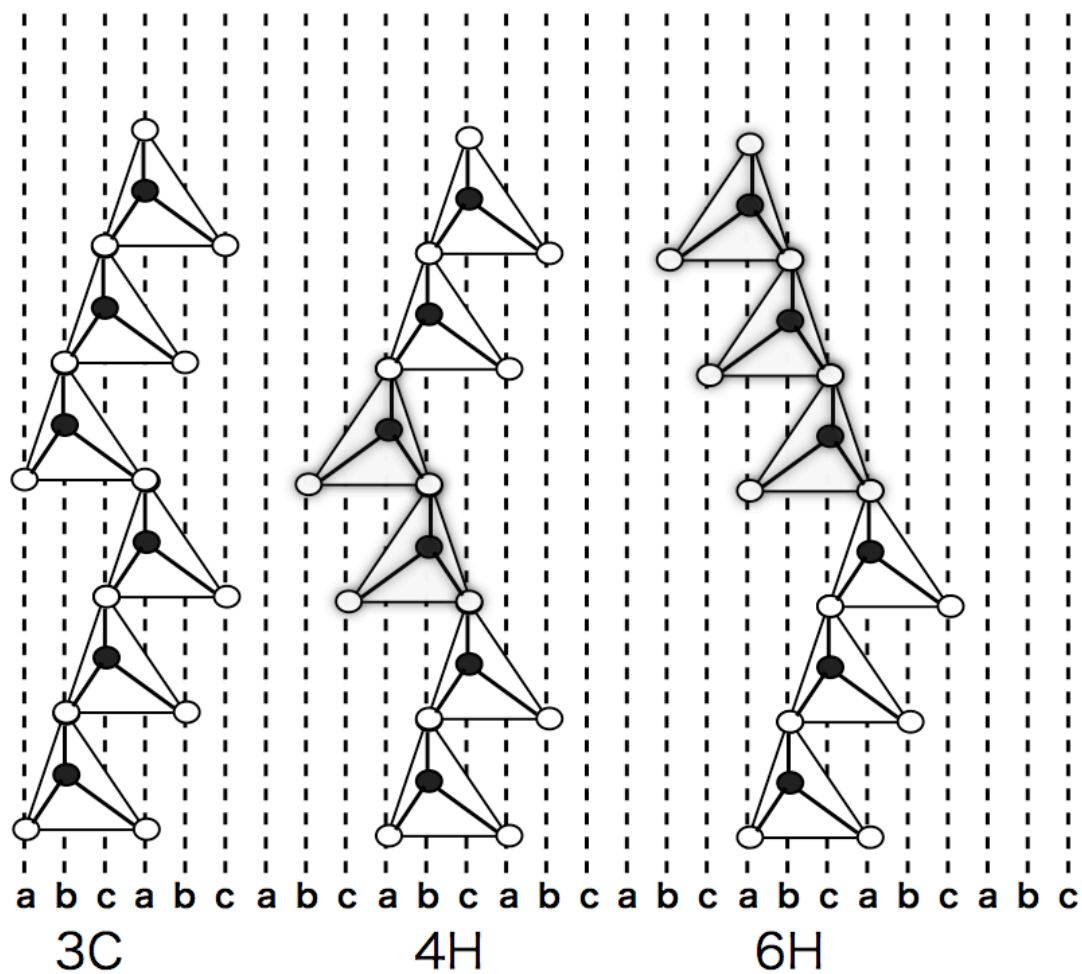


図 1.1: 代表的な SiC 結晶多形の結晶構造の $\{11\bar{2}0\}$ 面へ投影図． \blacktriangle がカーボン (C) , \triangle がシリコン (Si) である．中心に 1 つの Si , $1/4$ C を 4 つの頂点に配した正四面体を SiC の最小単位として , この正四面体の積層周期の違いが SiC における結晶多形構造である．

1.3 MSE

1.3.1 構成

MSE は 4H-SiC と 3C-SiC の二つの SiC のウエハーが液体 Si の薄膜を挟み込んだ構成になっており，ある一定の温度 (1700) のまま結晶成長をさせる．その時，熱的に安定な 4H-SiC は基板 (seed) となり，不安定な 3C-SiC は原料板 (feed) となる．そして，実験結果は等温過程において feed 側は溶け出し，seed 側に SiC 単結晶が析出した．

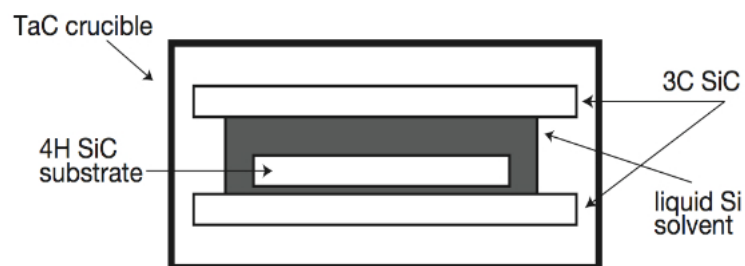


図 1.2: MSE の構成図．

1.3.2 駆動力

図 2.2 は MSE の駆動力を模式的に表した図である．結晶成長の際，MSE では温度は一定でありその時の駆動力は 3C-SiC と 4H-SiC の構造の違いから生じる化学ポテンシャル差となっている．そして，原料板である 3C-SiC から溶け出した C 原子が液体 Si を通して拡散し，基板である 4H-SiC に吸着することで SiC の単結晶が成長してゆく．



図 1.3: MSE の駆動力の模式図 .

1.3.3 成長機構

MSE は全く新しい成長法であり、その成長機構に関しては不明な点が多く面方位によって結晶成長の様子が異なる。また、MSE の SiC 単結晶成長においてその成長を律速する重要な機構が、表面とそれに直交する面が交差する「エッジ」である。そこで本研究は「エッジ」における原子レベルでの結晶成長の機構を理解することを目的とした。まず、3次元コンピュータグラフィックソフトウェアである Maya を用い、代表的な SiC 結晶である 3C, 4H, 6H-SiC の $\{0001\}$, $\{11-20\}$, $\{1-100\}$ 面が直交する格子モデルを作成し、SiC の「エッジ」を視覚化する。次に、VASP を用いて $\{11-20\}$, $\{1-100\}$ 面における原子付着の第一原理計算を行い、「エッジ」における結晶成長を考察する。

第2章 SiC表面の視覚化

2.1 SiCの構造を作成

2.1.1 Mayaについて

Maya はモデリング、アニメーション、レンダリングを統合的に扱える 3DCG ソフトウェアである。主に映画やゲームといったエンターテインメント分野向けの CG、とりわけフォトリアリスティックな「3DCG アニメーション」の作成を目的として、カナダのエイリアスシステムズが開発したソフトウェアである。また、MEL というスクリプト言語を用いて、Maya を自分の制作スタイルに合わせて自由にカスタマイズすることが可能である。

2.1.2 Rubyについて

Ruby はまつもとゆきひろ氏が開発したスクリプト言語である。Ruby は、強力なテキスト処理能力、シンプルな文法などの特長をもっている。インタプリタ型言語であるため、プログラムを作成したら、コンパイルなどの処理を行なうことなくすぐに実行することができる。また UNIX、MS-DOS、Windows、MacOS、BeOS の各プラットフォームに移植されている。

2.1.3 Ruby から MEL への変換

MEL は扱いにくい言語であるので、本研究では MEL よりもプログラミング環境の良い Ruby でメインスクリプトを作成した。メインスクリプト上で MEL ライブラリから必要なコマンドを呼び出し、MAYA インターフェイス上で出力する。

2.1.4 具体的な Ruby スクリプト

Stick & Ball で 4H-SiC のモデルを作成した時の Ruby スクリプトである．この Ruby スクリプトを見本としプログラムの説明を行う．

```
include Math
require 'pp'
require "matrix"

Dir.chdir("lib"){
  require 'input'
  require 'unitcell'
  require 'color'
  require 'make_cell'
  require 'cell_support'
  require 'currentTime'
  require 'object'
  require 'light'
  require 'camera'
}
```

ここで，lib ディレクトリに入っている Ruby スクリプトを関数として呼び出している．

```
data_name = "4H-SiC.txt"
filename = "4H-SiC_corner.mel"
```

ここで読み込むテキストデータと出力される MEL ファイルの名前を指定する．

```
unitcell,ball,p_vec,name = input(data_name)
```

ここで，テキストデータの中のユニットセルのベクトル，プリミティブベクトル等のデータの読み込みを行う．

```
kind = name.size
pp ball

ball1 = {}
```

```

cell_scale = [[-3,3],[-0,0],[-3,3]]
for i in 0..kind-1 do
  ball1[name[i]] = unitcell_basis(ball[name[i]],p_vec,cell_scale)
end

cell_scale = [[-3,3],[1,1],[-3,3]]
tmp = {}
for i in 0..kind-1 do
  tmp[name[i]] = unitcell_basis(ball[name[i]],p_vec,cell_scale)
end

for i in 0..kind-1 do
  for j in 0..tmp[name[i]].size-1 do
    ball1[name[i]].push tmp[name[i]][j]
  end
end

def layer_add(in_ball,out_ball,p_vec,cell_scale,name,layer)
  tmp = []
  tmp.push in_ball[name][layer]
  tmp = unitcell_basis(tmp,p_vec,cell_scale)

  for i in 0..tmp.size-1 do
    flag = 0
    out_ball[name].each do |val|
      if(val==tmp[i])then flag=1 end
    end
    if(flag!=1)then
      out_ball[name].push tmp[i]
    end
  end

  return out_ball
end

```



```

cell_scale = [[-3,3],[2,2],[-3,3]]
ball1 = layer_add(ball,ball1,p_vec,cell_scale,name[0],0)
cell_scale = [[-3,3],[1,1],[-3,3]]
ball1 = layer_add(ball,ball1,p_vec,cell_scale,name[0],0)

```

原子の入力やセルの拡張等を行っている .

```

for i in 0..kind-1 do
  j = 0
  while(j<ball1[name[i]].size)do
    x = ball1[name[i]][j][0]
    y = ball1[name[i]][j][1]
    z = ball1[name[i]][j][2]

    if((x>4.622 || x<-4.622)&&y>10.86)then
      ball1[name[i]].delete_at(j)
    elsif((z>5.337 || z<-5.337)&&y>10.86)then
      ball1[name[i]].delete_at(j)
    else
      j += 1
    end
  end
end
end

```

エッジを表示させるために , 表示させる原子を限定させている .

```
##### MAKE_MEL
```

以下が MEL ファイルの出力データである .

```

Dir.chdir("mel"){
  file1 = File.open(filename,'w')

  color = []
  file1,color[0] = color_lambert(file1,[0.7,0.7,1])
  file1 = color_incandescence(file1,color[0],[0.2,0.2,0.8])
  file1,color[1] = color_lambert(file1,[0.3,0.3,1])#Si

```

```

file1,color[2] = color_lambert(file1,[0.4,0.4,0.4])#C
file1,color[3] = color_lambert(file1,[0.8,0.8,0.8])
file1,color[4] = color_lambert(file1,[1,0,0])
file1,color[5] = color_lambert(file1,[0,1,1])
file1 = color_transparency(file1,color[5],[0.6,0.6,0.6])
file1,color[6] = color_lambert(file1,[1,1,0])
file1 = color_transparency(file1,color[6],[0.6,0.6,0.6])
file1,color[7] = color_lambert(file1,[1,0,1])
file1 = color_transparency(file1,color[7],[0.6,0.6,0.6])

```

ここで、色の指定を行っている。

```

file1 = object_add(file1,'sphere','background',color[0])
file1 = currentTime_move(file1,0,'background',[0,0,0])
file1 = currentTime_scale(file1,0,'background',[300,300,300])
file1 = currentTime_rotate(file1,0,'background',[0,0,0])

scale = [0.5,0.5,0.5]
for i in 0..kind-1 do
    file1 = make_ball_cell(file1,ball1[name[i]],color[i+1],name[i],scale)
end
scale = [1,0.1,0.1]
file1 = make_stick_cell(file1,ball1[name[0]],ball1[name[1]],[0,2.2],
    color[3],scale)

```

ここで、Stick と Ball を作成している。

```

file1 = object_add(file1,'nurbsCube','face1',color[5]) # {0001}
file1 = currentTime_move(file1,0,'face1',[0,10.085,0])
file1 = currentTime_scale(file1,0,'face1',[30,0,30])
file1 = currentTime_rotate(file1,0,'face1',[0,0,0])

file1 = object_add(file1,'nurbsCube','face2',color[6]) # {11-20}
file1 = currentTime_move(file1,0,'face2',[4.621,(20.17*3)/4,0])
file1 = currentTime_scale(file1,0,'face2',[10.085,0,5.336*2])
file1 = currentTime_rotate(file1,0,'face2',[0,0,90])

file1 = object_add(file1,'nurbsCube','face3',color[7]) # {1-100}

```

```

file1 = currentTime_move(file1,0,'face3',[0,(20.17*3)/4,5.336])
file1 = currentTime_scale(file1,0,'face3',[4.621*2,0,10.085])
file1 = currentTime_rotate(file1,0,'face3',[90,0,0])

file1 = wire_frame_unitcell(file1,p_vec,color[4])
file1.close
}

```

ここで (0001) , (11-20) , (1-100) 面を表示させている .

```
puts 'output >>> '+filename
```

書き出し終了 .

2.1.5 SiC の 3DCG モデル

Maya を用いて作成した 4H , 6H , 3C-SiC の 3DCG モデルである (図 2.1)(図 2.2)(図 2.3) . Si 原子は白で表し , C 原子は黒で表した . また , {0001} , {11-20} , {1-100} 面で区切り , 結晶表面のエッジを表示させた . {0001} 面は緑 , {11-20} 面は灰 , {1-100} 面は紫で表している . これにより SiC 表面がどのような構造をとっているのかが一目で分かるようになった .

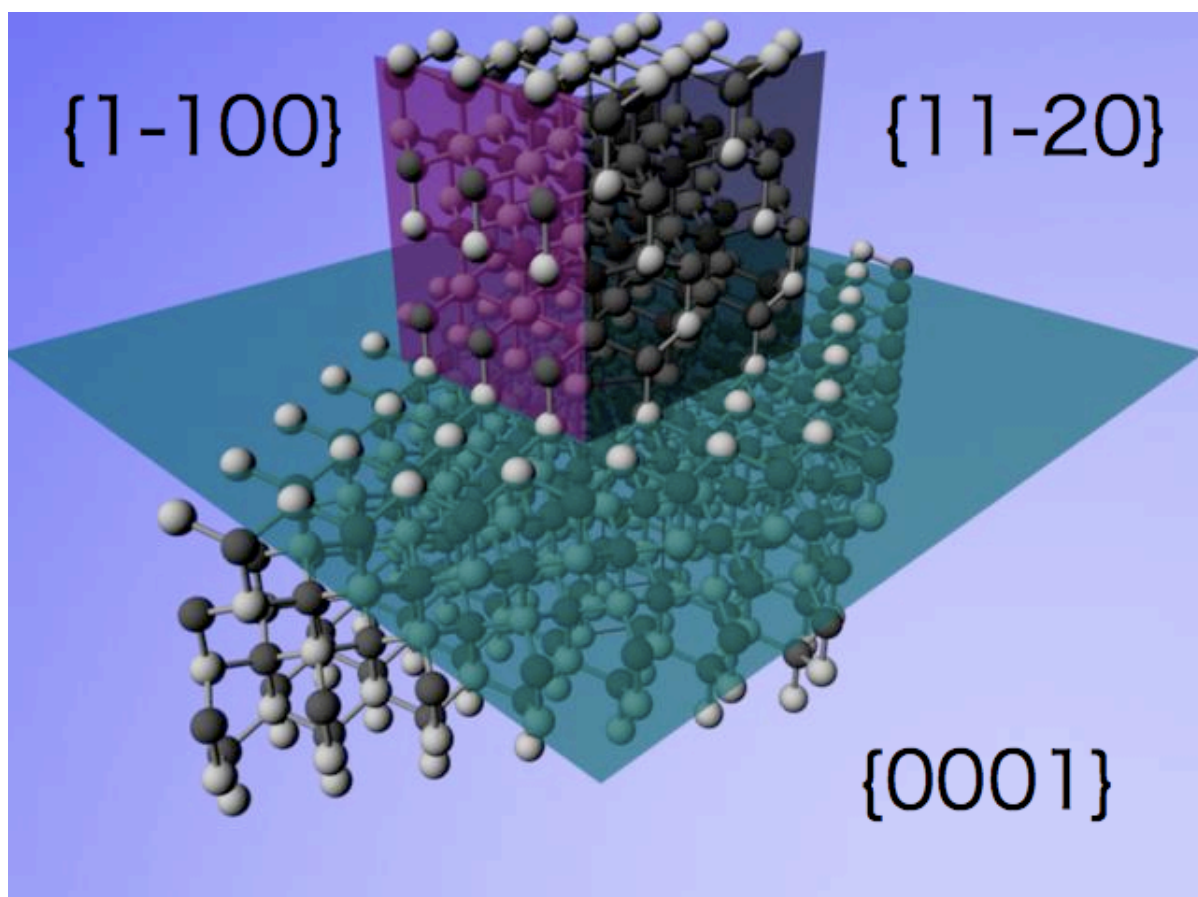


図 2.1: 4H-SiC のモデル .

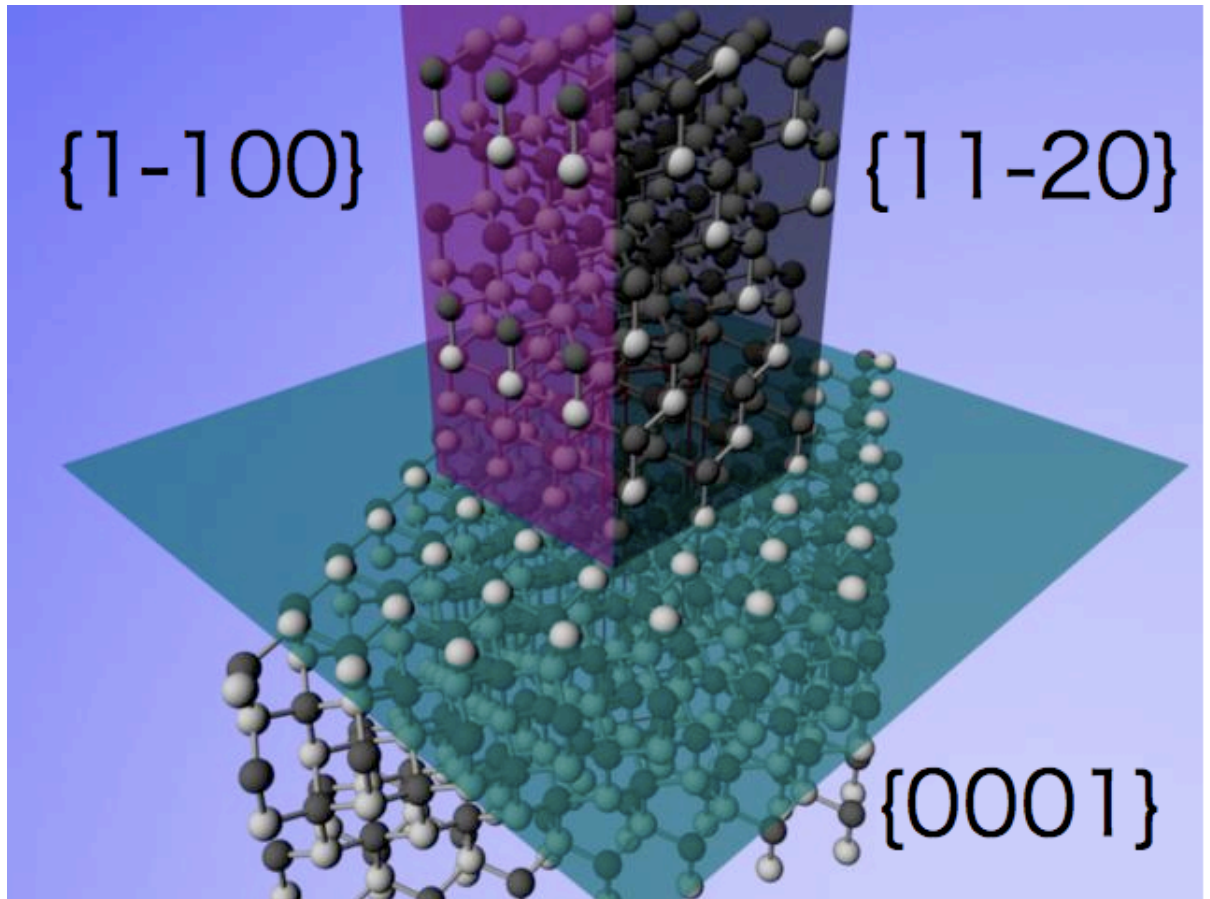


図 2.2: 6H-SiC のモデル .

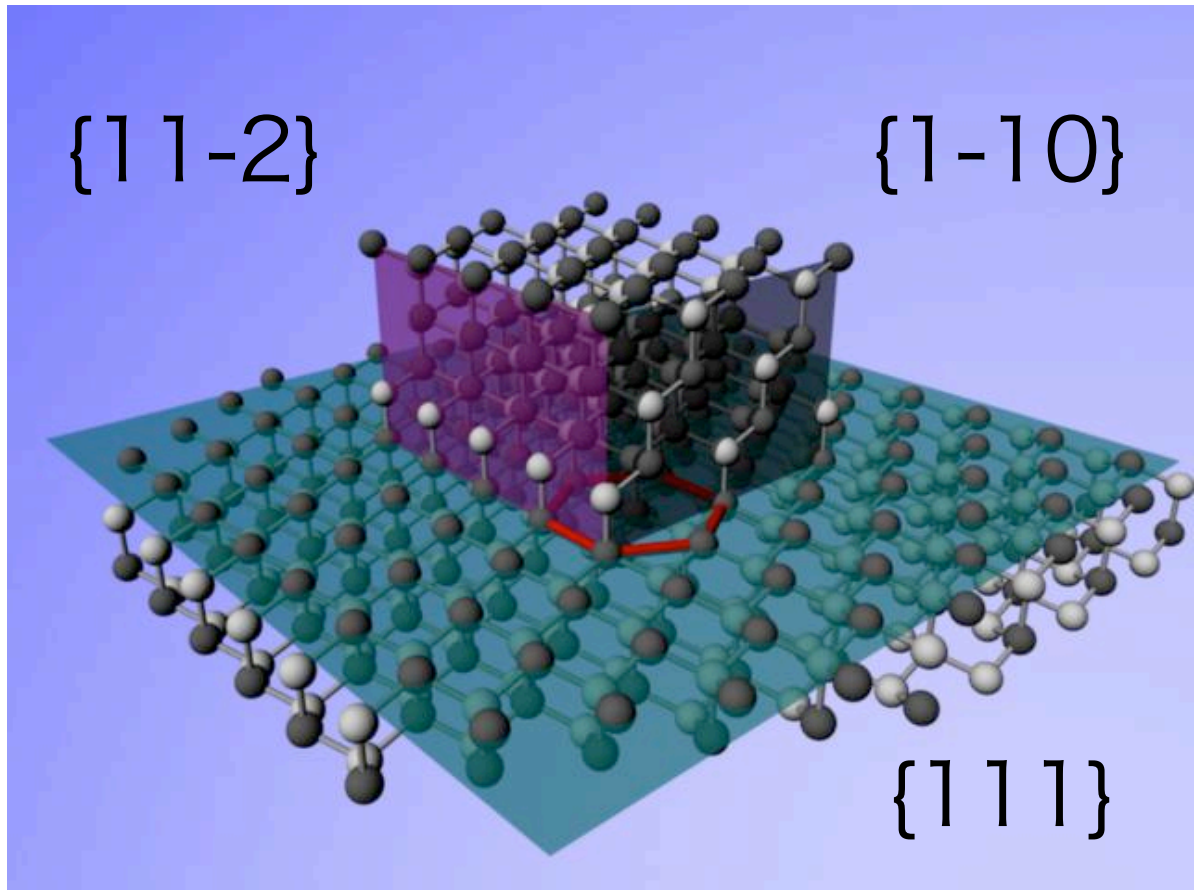


図 2.3: 3C-SiC のモデル .

2.1.6 SiC エッジの 3DCG アニメーション

Maya を用いて作成した 4H-SiC のエッジの傾斜を表したアニメーションである (図 2.4) ~ (図 2.10) . 4H , 6H , 3C-SiC の 3DCG モデルと同様に Si 原子は白で表し , C 原子は黒で表した . また , 拡散する C 原子を黄色で表した . このアニメーションでは $\{11-20\}$ と $\{1-100\}$ に段差を作り , 結晶成長の途中経過を表した . アニメーション 1 は $\{11-20\}$ 面を横から見た図であり , アニメーション 2 は $\{1-100\}$ 面を横から見た図である . これらの図から見て分かるように , 4H-SiC は $\{1-100\}$ 面より $\{11-20\}$ 面の方が

傾斜が急である，したがって， $\{1-100\}$ 面より $\{11-20\}$ 面の方が結晶成長の早いことが予想できる．

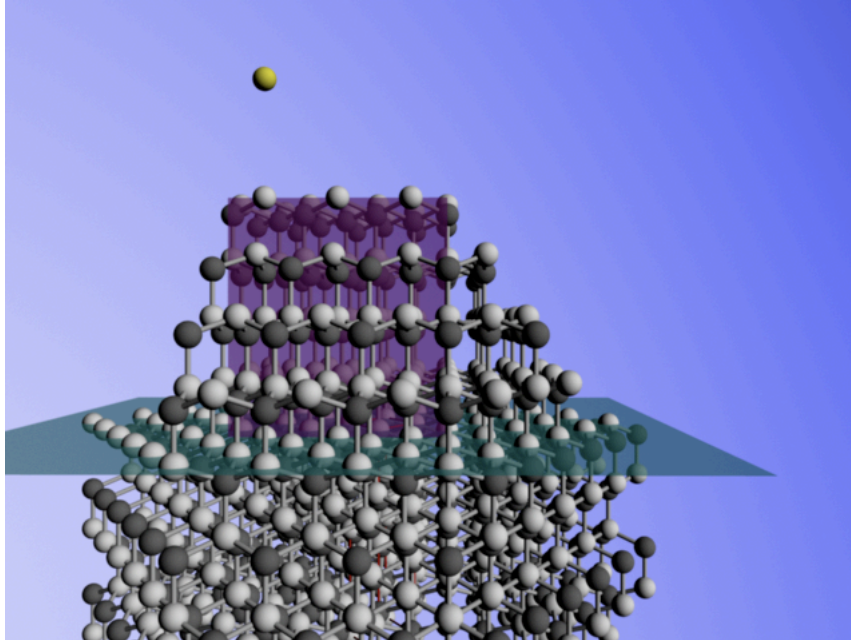


図 2.4: アニメーション 1 .

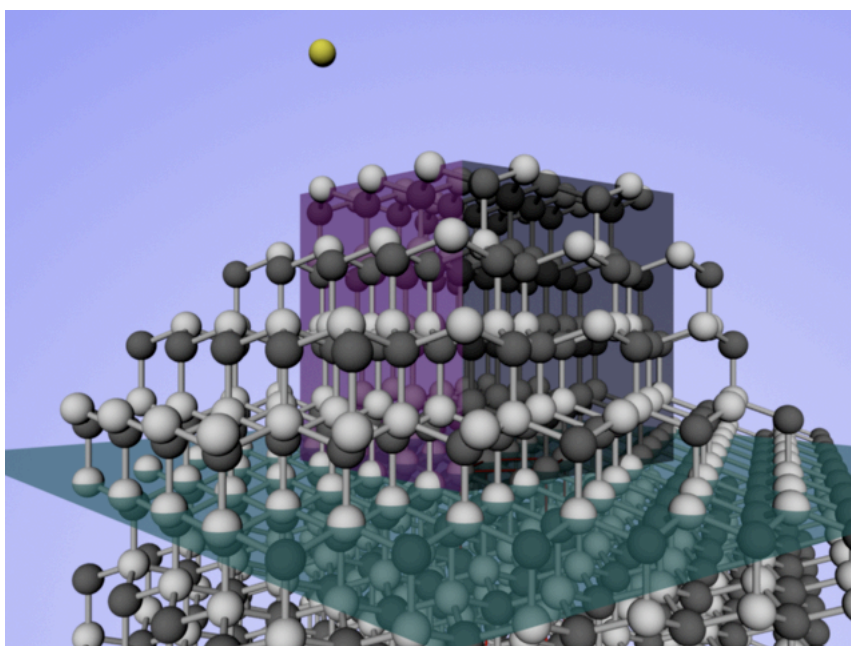


図 2.5: アニメーション 2 .

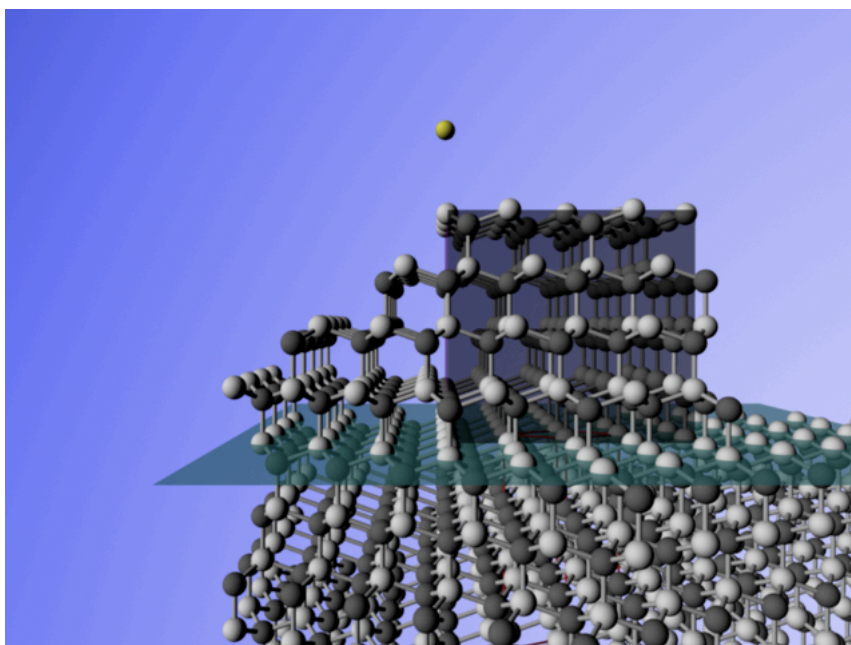


図 2.6: アニメーション 3 .

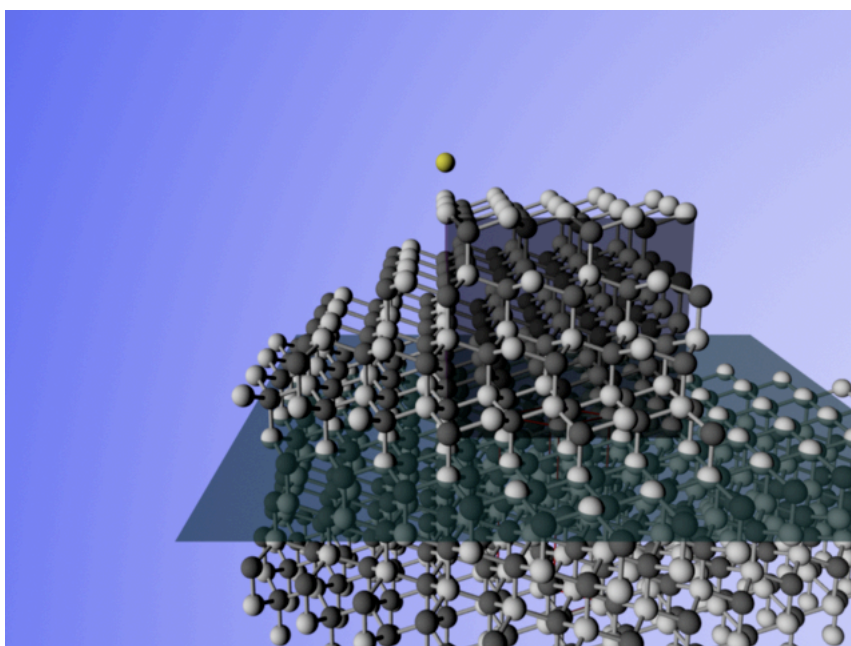


図 2.7: アニメーション 4 .

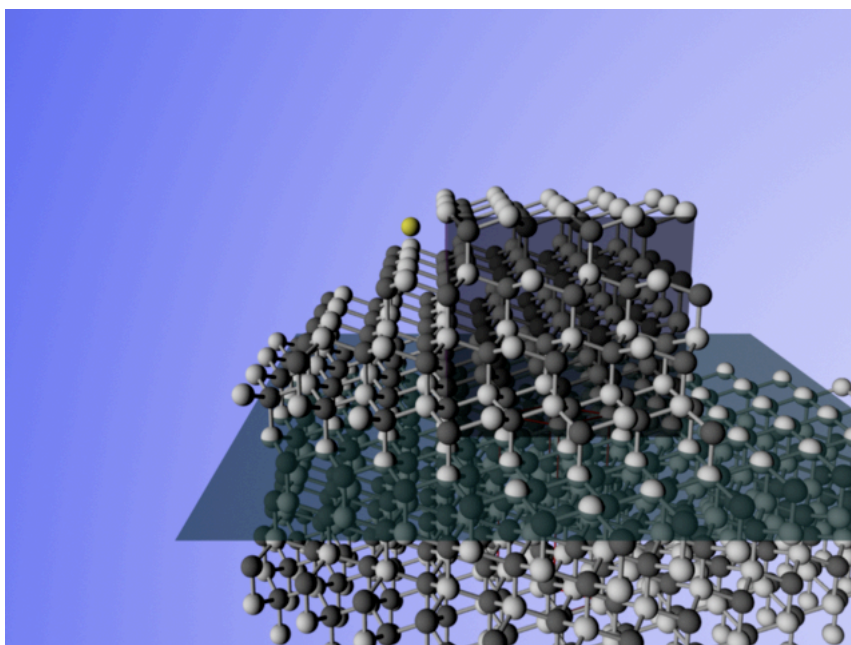


図 2.8: アニメーション 5 .

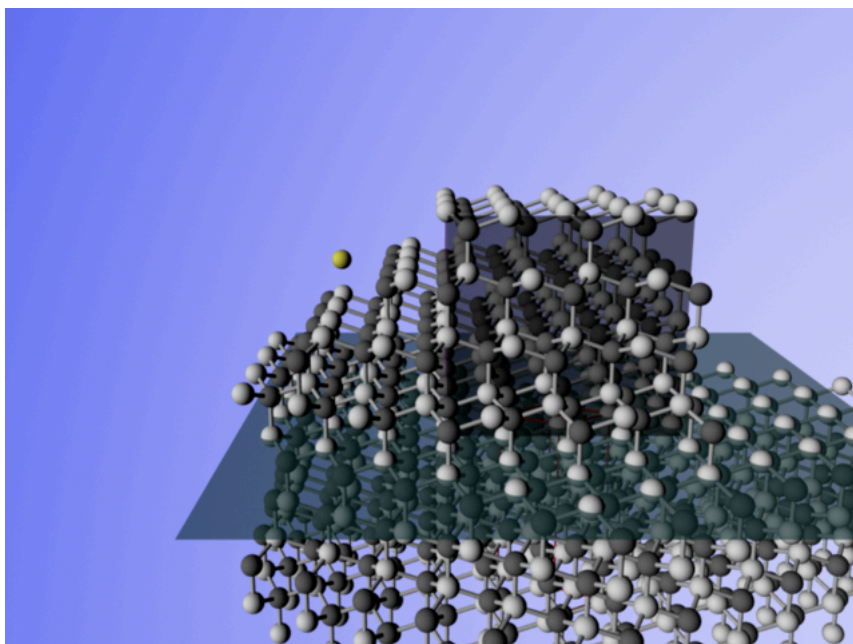


図 2.9: アニメーション 6 .

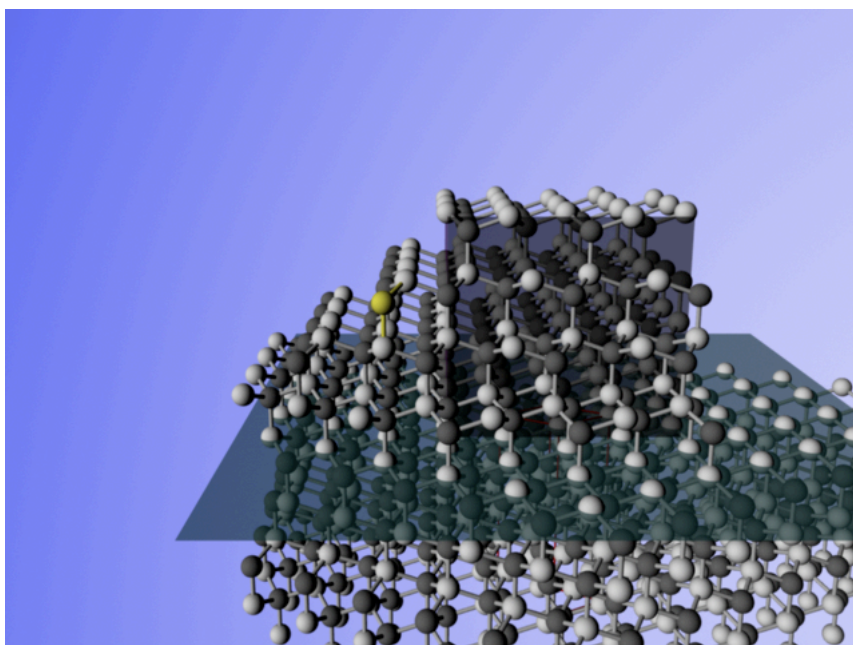


図 2.10: アニメーション 7 .

第3章 SiC表面での原子付着の 第一原理計算

3.1 計算原理

今回、4H-SiCにおける $\{11\cdot20\}$ 、 $\{1\cdot100\}$ 面の結晶表面でいくつかの位置にC原子を付着させて第一原理計算をした。原子の移動経路を導き出すために、4H-SiCにおける $\{11\cdot20\}$ 、 $\{1\cdot100\}$ 面のスラブモデルの結晶表面にC原子を1つ付着させ、そのC原子のエネルギーを計算し、その結果から表面拡散時のC原子の移動経路を考察する。

3.2 計算手法

C原子のエネルギーを求める計算の流れについて簡単に説明する。

1. 原子モデル構築ソフト Medea を用いて 4H-SiC のバルクモデルと $\{11\cdot20\}$ 、 $\{1\cdot100\}$ 面のスラブモデルを作成する。
2. 各面のスラブモデルに C 原子を 1 つ付着させる。
3. VASP を用いて第一原理計算を行い、2 で作成した C 原子を 1 つ付着させたモデルのエネルギーを求める。
4. 3 で求めた C 原子を 1 つ付着させたモデルのエネルギーと付着させる前のエネルギーの差分を取り、C 原子一つあたりのエネルギーを求める。
5. 計算結果から表面拡散時の原子の移動経路を考察する。

3.2.1 MedeA について

MedeA はデータベースと第一原理計算の手法を一つのプラットフォームで統合した材料設計支援のための統合ソフトウェアである．グラフィックインターフェースおよび計算プログラムは全て Windows システム上で稼働するので，構造の検索，構築，編集，計算，解析までを一つのプラットフォーム上で行うことができる．MedeA を使って 4H-SiC のバルクモデルと $\{11\cdot20\}$ ， $\{1\cdot100\}$ 面のスラブモデルを作成した．

3.2.2 C 原子を付着させる手順

ターミナル上で `setup 1.0` というコマンドを入力すると，MedeA で作成した原子モデルのファイルの中に `POSCAR.orig` というファイルが作成される．このファイルの中には計算する原子モデルの格子定数，原子数，原子位置が収められている．C 原子を 1 つ結合させるにはファイルの中に C 原子の原子位置を書き加え，さらに C 原子の原子数を 1 つ分足しておく．また，C 原子の原子位置は Maya で確認しながら位置を決定した．

3.2.3 非平面派基底擬ポテンシャル法

密度汎関数に基づく第一原理擬ポテンシャル法によるバンド計算 VASP code を用いて計算を行う．この手法では，3 次元周期境界条件を満たす平面波の基底関数を用いて電子被占有の軌道を展開し，その波動関数をもとに一電子方程式を解くことにより電子状態を求める．交換相関ポテンシャルは GGA (generalized gradient approximation) を用いた．そして，擬ポテンシャルとして PAW (projector augmented wave) を用いた．また，ターミナル上で `allqsub` というコマンドを入力すると計算が実行される．

3.2.4 C 原子の付着位置

C 原子は $\{11\cdot20\}$ 面では図 3.1 の 1～6 点， $\{1\cdot100\}$ 面では図 3.2 の 1～12 点に置いた．これらの点に置くことにより，他の点はすべて等価な点とみなすことができるので，C 原子が付着すると予想される全ての点に置

いたこととなる．そして， z 軸方向にリラクゼーションさせ，エネルギー計算を VASP で行った．

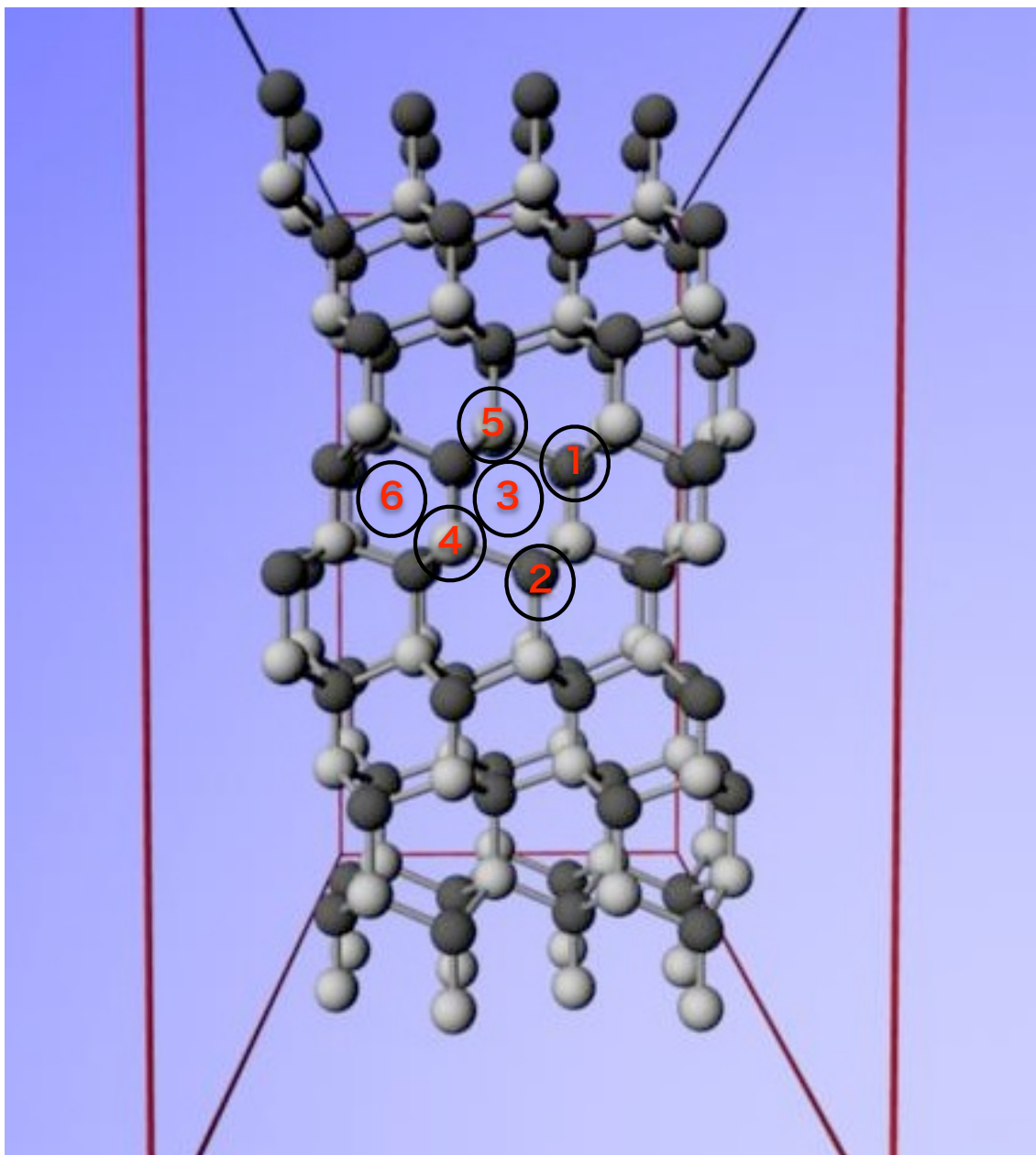


図 3.1: 4H-SiC の $\{11-20\}$ 面における C 原子の付着位置．

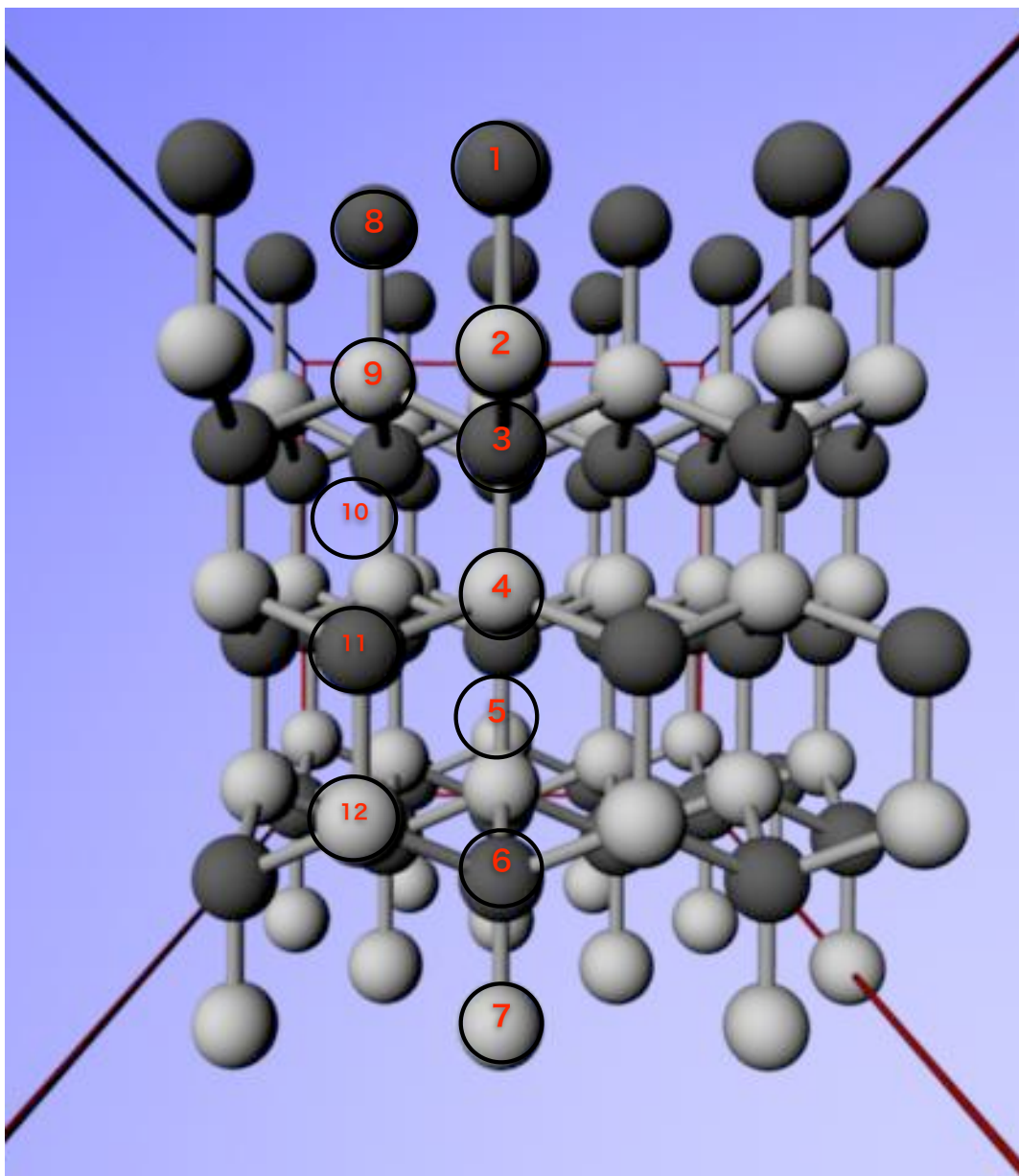


図 3.2: 4H-SiC の $\{1-100\}$ 面における C 原子の付着位置 .

3.3 計算結果の考察

3.3.1 計算結果 (最安定エネルギー)

表 3.1: {11-20} 面における最安定エネルギー .

	site	最安定エネルギー	C 原子 1 つあたりのエネルギー
4H-SiC{11-20}	6	-869.80995	-6.97070
	5	-869.37393	-6.53468
	2	-868.50398	-5.66473
	3	-868.29537	-5.45612
	4	-867.02376	-4.18451
	1	-867.00121	-4.16196

表 3.2: {1-100} 面における最安定エネルギー .

	site	最安定エネルギー	C 原子 1 つあたりのエネルギー
4H-SiC{1-100}	8	-673.20321	-10.90153
	7	-671.01664	-8.71496
	1	-670.85822	-8.55654
	9	-669.54070	-7.23902
	6	-668.77603	-6.47435
	10	-668.74773	-6.44605
	11	-668.65450	-6.35282
	4	-667.19760	-4.89592
	2	-666.95340	-4.65172
	3	-666.83958	-4.5379
	12	-666.25418	-3.9525
	5	-666.11210	-3.81042

3.3.2 計算結果 (最安定位置)

図 3.3 ~ 図 3.6 の黄色で表されたボールは VASP で計算された後の C 原子である．Maya を用い計算後の C 原子を一度に表示させた．また表 3.1 , 表 3.2 より最安定位置な site6 , site8 のみ赤色のボールで表した．

表 3.3: $\{11\text{-}20\}$ 面における最安定位置．

site	最安定位置 (x)	最安定位置 (y)	最安定位置 (z)
1	6.66800	16.84358	11.75800
2	5.77900	16.75727	9.42200
3	5.33500	16.41909	10.99200
4	4.00104	17.21161	10.05000
5	4.89000	16.04214	12.56200
6	2.66900	16.22943	10.99200

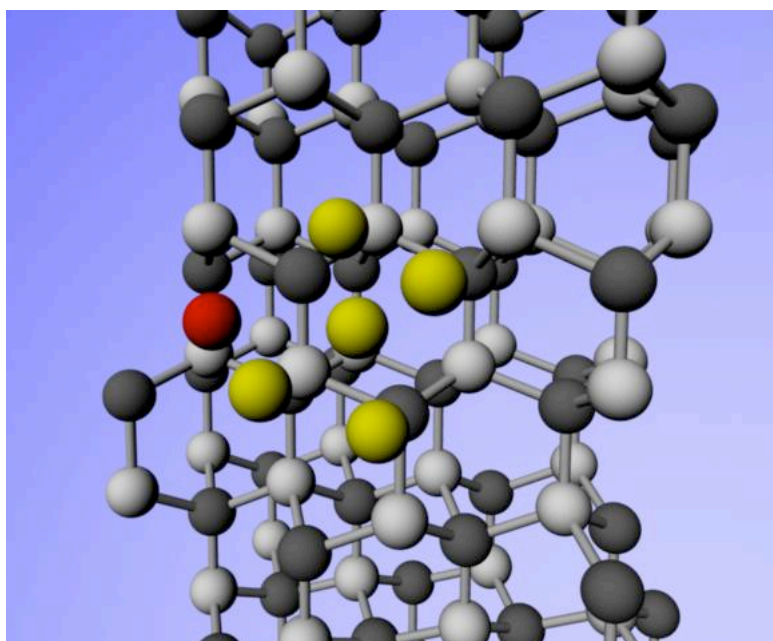


図 3.3: $\{11\text{-}20\}$ 面における最安定位置．

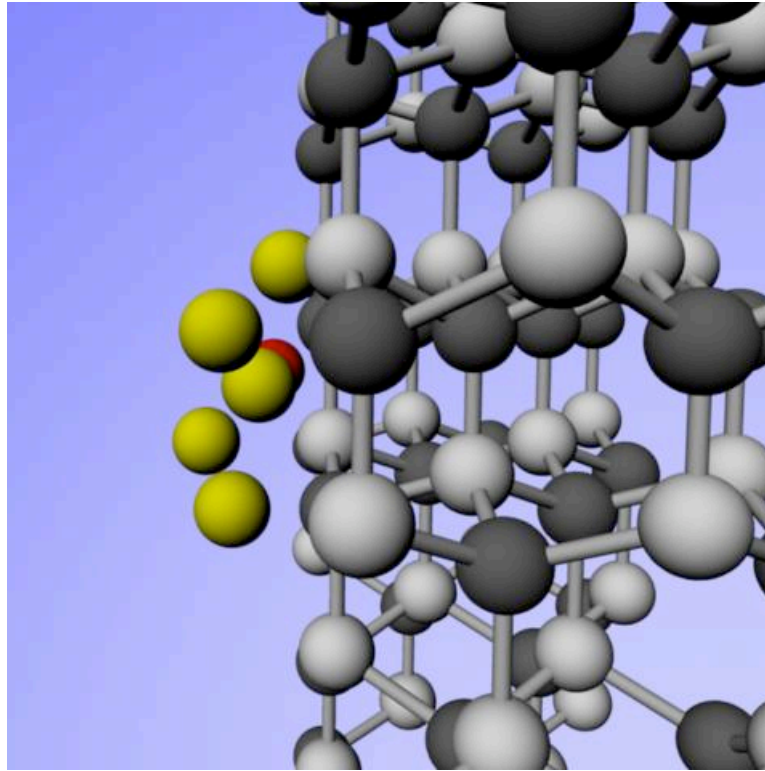


図 3.4: 図 3.3 を横から見た図 .

表 3.4: $\{1-100\}$ 面における最安定位置 .

site	最安定位置 (x)	最安定位置 (y)	最安定位置 (z)
1	0.62813	4.62000	21.06797
2	2.51250	4.62000	21.58273
3	3.31650	4.62000	21.42338
4	5.02500	4.62000	20.12845
5	6.59500	4.62000	20.13289
6	8.34150	4.62000	19.71220
7	10.05000	4.62000	20.90641
8	0.62813	6.16000	19.85639
9	2.51300	6.16000	20.67799
10	4.17100	6.16000	20.00282
11	5.65313	6.16000	20.25066
12	7.53750	6.16000	20.75683

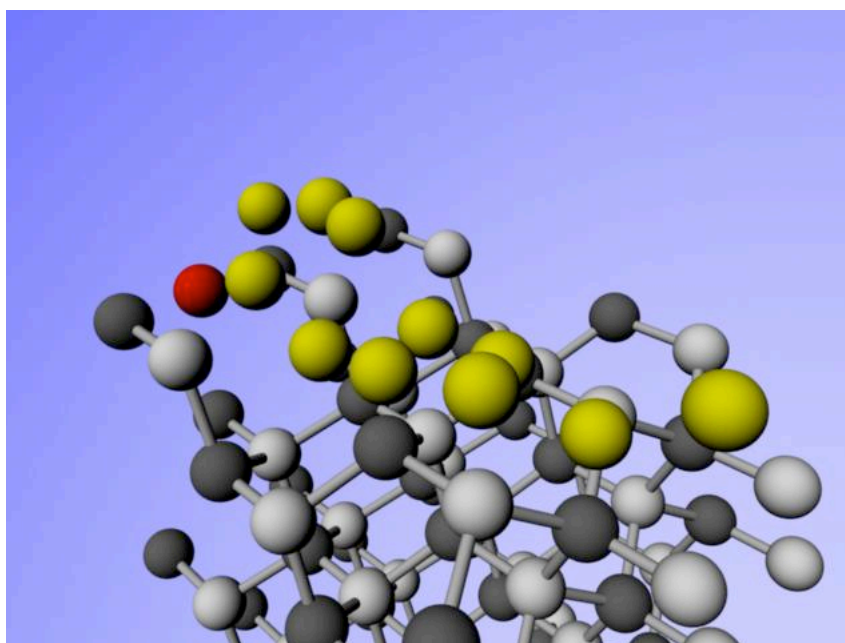


図 3.5: $\{1-100\}$ 面における最安定位置 .

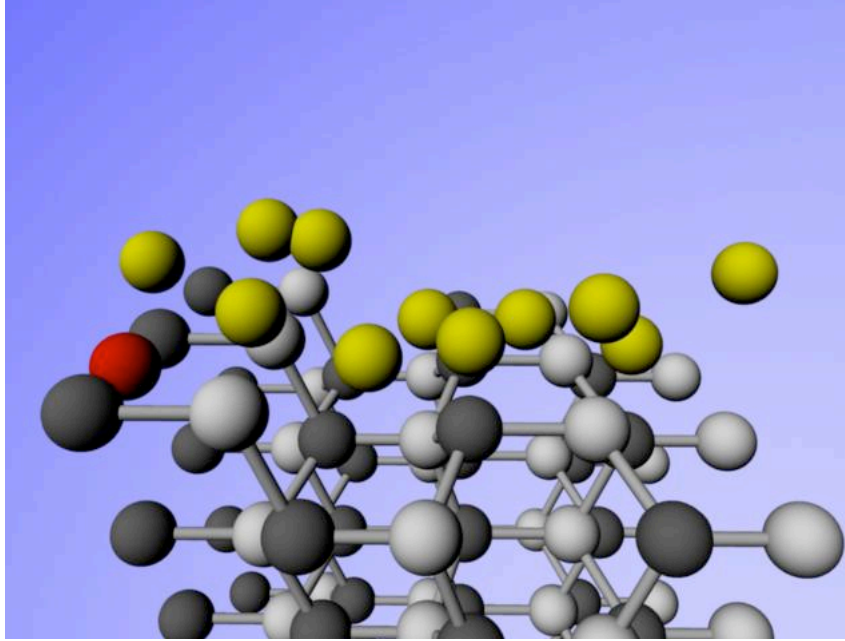


図 3.6: 図 3.5 を横から見た図 .

3.3.3 考察

{11-20} 面において拡散してきた C 原子は最安定な site6(表 3.1), またはそれと等価な site から付着してゆき, {1-100} 面においては最安定な site8(表 3.2), またはそれと等価な site から付着してゆくことが分かった . また, {11-20}, {1-100} 面共にそれぞれの C 原子付着点におけるエネルギーに大きな差がある結果となった (表 3.1)(表 3.2) . これは {11-20}, {1-100} 面において C 原子の拡散が行われにくいことを表している . MSE の実験結果では {11-20}, {1-100} 面に比べ {0001} 面に顕著な成長が現れているが (図 3.7), これは {11-20}, {1-100} 面においてエネルギーバリアが大きく, C 原子の拡散が行われにくいことが原因となっていることが予想される . また, {1-100} 面は {11-20} 面に比べエネルギーバリアが大きく, {11-20} 面の方が C 原子の拡散はより行われにくく, 成長も遅いということが分かる . これらをふまえて {11-20}, {1-100} 面の成長を考えると, エネルギーバリアが大きい {11-20}, {1-100} 面に {0001} 面から C 原子が拡散することはほとんどなく, {11-20}, {1-100} 面と {0001} 面の

「エッジ」が「つらら」の様な形で成長してゆくのではないかと予想できる (図 3.8) . また , $\{11-20\}$ 面よりエネルギーバリアが大きい $\{1-100\}$ 面の方がより鋭角なつららを形成することも予想できる .

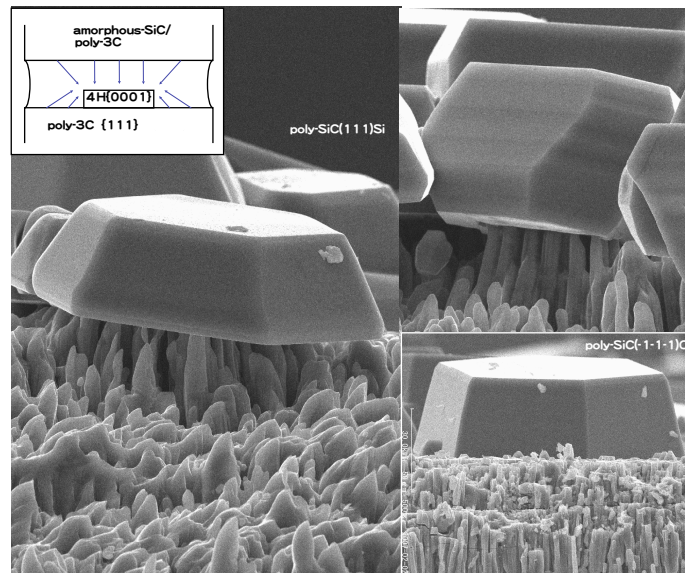


図 3.7: MSE の実験結果 .

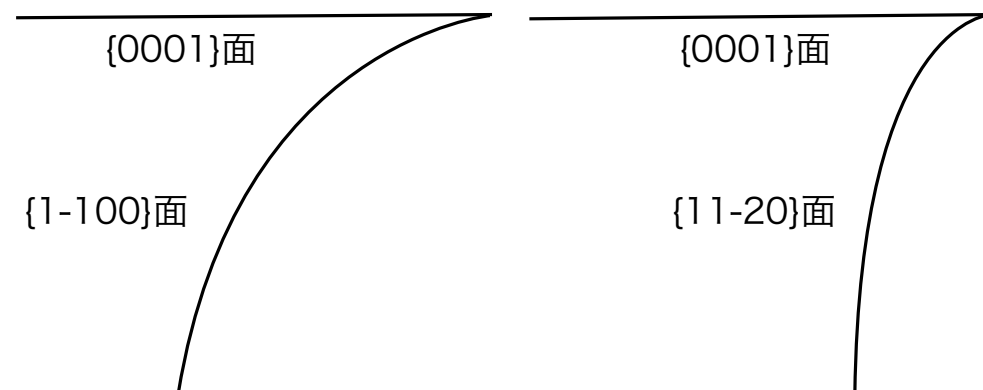


図 3.8: $\{1-100\}$, $\{11-20\}$ 面の成長の模式図 .

第4章 総括

本研究では SiC の「エッジ」における原子レベルでの構造をモデル化し、4H-SiC における $\{11-20\}$ 、 $\{1-100\}$ 面の結晶表面でのいくつかの位置に原子を付着させ第一原理計算を行った。その結果、以下のような知見が得られた。

$\{11-20\}$ 、 $\{1-100\}$ 面における C 原子付着位置の最安定な場所が求まった。また、 $\{11-20\}$ 、 $\{1-100\}$ 面共にエネルギーバリアが大きく、C 原子の拡散が行われにくいことが分かった。つまり、エネルギーバリアが大きい $\{11-20\}$ 、 $\{1-100\}$ 面に $\{0001\}$ 面から C 原子が拡散することはほとんどなく、 $\{11-20\}$ 、 $\{1-100\}$ 面と $\{0001\}$ 面の「エッジ」が「つらら」の様な形で成長してゆくのではないかと予想できる。なお、 $\{11-20\}$ 面よりエネルギーバリアが大きい $\{1-100\}$ 面の方がより鋭角なつららを形成することも予想できる。

謝辞

本研究を遂行するにあたり，終始多大なる有益なご指導，及び丁寧な助言を頂いた西谷滋人教授，戸賀瀬健介さんに深い感謝の意を表します．

また，本研究を進めるにつれ，西谷研究室員の皆様にも様々な知識の供給，ご協力を頂き，本研究を大成する事ができました．最後になりましたが，この場を借りて心から深く御礼申し上げます．

参考文献

- [1] 関西学院大学，博士論文 2004，pp.1-17，浅岡康：『極薄液層環境を用いた単結晶 SiC 薄膜成長に関する研究』．
- [2] 『炭化珪素の特徴』 <http://www.geocities.co.jp/Bookend-Kenji/5046/sic2.html> (2009/2/10 アクセス) ．
- [3] 荒井和雄，吉田貞史 共著『SiC 素子の基礎と応用』(オーム社，2003) ．
- [4] 上羽牧夫，責任編集『結晶成長のしくみを探る その物理的基礎』(共立出版株式会社，2002) ．
- [5] 林田，橋口智仁，konkon，黒田あや子，本城なお．『AUTODESK MAYA』 Maya で始める 3DCG 制作の基本 (株式会社ワークスコーポレーション 2006) ．
- [6] 高橋征義，後藤裕蔵 共著，まつもとゆきひろ 監修 『たのしい Ruby』 ．