

# 卒業論文

## チタンの面欠陥エネルギーの第一原理計算

関西学院大学 理工学部 情報科学科

2681 真田 裕示

2006年 3月

指導教員 西谷 滋人 教授

## 概 要

本研究では、チタン (以下 Ti) の面欠陥エネルギーを第一原理計算によって求め、2 体間ポテンシャルの結果の  $\gamma$ -surface と比較した。  $\gamma$ -surface とは面状の欠陥に伴うエネルギー変化を表現する図として知られている。これは層状に積んだ格子模型の一面を境にその上下の格子を特定の方向にずらす。そのずれに伴ってどのようなエネルギー変化をなすかをプロットした図である。これによってどのような面状のずれが起こり易いかが一目で分かる。現在までに、2 体間ポテンシャルで Ti の面欠陥エネルギーの  $\gamma$ -surface は表現されてきた。しかし、2 体間ポテンシャルで表される hcp-Ti の  $\gamma$ -surface は、現実の hcp-Ti の変形挙動からは大きくはずれた結果であり、F(stable fault) 点は本来なら 2 体間ポテンシャルでの結果の位置よりも高い位置にある事が予測されている。

そこで本研究では、原子の種類だけから電子構造を求め、色々な物性を予測する。第一原理計算を用いて面欠陥エネルギーを計算し、  $\gamma$ -surface を再現して 2 体間ポテンシャルの結果と比較する。Ti を初め、金属は曲げたり引っ張ったりする事によって形が変わる。変形にはすべり変形と双晶変形の 2 種類がある。すべり変形は、単結晶では特定の面に沿って特定の方向へすべる。その結果、試料の表面にすべり線あるいはすべり帯と呼ばれるステップが生じる。また、多結晶体では、各結晶粒内で複数のすべりが同時に活動することによって、結晶粒界で連続性を保ちながら変形が進行する。双晶変形は、結晶格子がある領域にわたって双晶面に平行に一様なせん断変形を起こして、その部分が母格子と双晶関係をもつ方位に変換する現象である。

Ti は双晶変形もすべり変形も両方する事が知られているが、本研究では双晶変形した Ti を研究課題とした。双晶変形が起こると原子のならびがズレ、積層欠陥が生じる。積層欠陥とは、結晶の原子配列を特定の結晶面に平行な原子面が規則的に積み重なった状態であるとき、その積み重なり方に生じた狂いのことである。そこで、結合エネルギーを第一原理計算によって求め、積層欠陥の起こりやすさを調べる。本研究の第一原理計算による  $\gamma$ -surface では F 点がもっとも高い欠陥エネルギー値の 3 分の 2 程度であり、本来の予測通りの結果が得られた。

# 目次

第1章	目的	3
1.1	研究目的と背景	3
1.1.1	研究目的	3
1.1.2	背景	4
第2章	計算手法	5
2.1	unit cell での Ti の第一原理計算の精度の検証	5
2.1.1	それぞれの立体構造	5
2.1.2	計算精度の高い FP-LMTO 法	7
2.1.3	VASP での計算結果	9
2.1.4	まとめ	13
2.2	縦に積んだ時の計算精度の検証	13
2.2.1	unit cell を縦に積む方法	13
2.2.2	unit cell が増えた時の計算時間と精度	16
2.2.3	unit cell を縦に積んだ時の計算精度の結果	18
2.3	まとめ	22
第3章	結果及び考察	24
3.1	面欠陥エネルギーとは	24
3.1.1	結晶の一部をずらす方法	24
3.2	構造緩和	26
3.2.1	内部緩和	26
3.2.2	外部緩和	28
3.3	失敗例	31
3.4	計算結果	33
3.4.1	2 体間 potential での $\gamma$ -surface	34
3.4.2	第一原理計算での面欠陥エネルギーの $\gamma$ -surface	35
第4章	まとめ	37
付録A	エネルギー計算の効率化	39
A.1	効率化の必要性	39
A.2	プログラム	39

A.3 原子ポテンシャルファイルの変更 . . . . .	39
付 録 B 結合エネルギー曲線の描画方法	41
B.1 グラフの書き方 . . . . .	41
B.2 Maple でのプログラム . . . . .	41
付 録 C 境界面の変換プログラム	44

# 第1章 目的

## 1.1 研究目的と背景

### 1.1.1 研究目的

チタン (以下 Ti) は、軽量で高剛性の金属であることから、工学の各分野に応用されている。Ti を初め、金属は曲げたり引っ張ったりする事によって形が変わる。変形にはすべり変形と双晶変形の2種類がある。

すべり変形は、単結晶では特定の面に沿って特定の方向へすべる。その結果、試料の表面にすべり線あるいはすべり帯と呼ばれるステップが生じる。また、多結晶では、各結晶粒内で複数のすべりが同時に活動することによって、結晶粒界で連続性を保ちながら変形が進行する。双晶変形は、結晶格子がある領域にわたって双晶面に平行に一様なせん断変形を起こして、その部分が母格子と双晶関係をもつ方位に変換する現象である。

Ti は双晶変形もすべり変形も両方する事が知られているが、本研究では双晶変形した Ti を研究課題としている。双晶変形が起こると原子の並びがズレ、積層欠陥が生じる。積層欠陥とは、結晶の原子配列を特定の結晶面に平行な原子面が規則的に積み重なった状態であると見るとき、その積み重なり方に生じた狂いのことである。そこで、結合エネルギーを第一原理計算によって求め、積層欠陥の起こりやすさを調べる [1]。

### 1.1.2 背景

現在までに、2体間ポテンシャルでTiの面欠陥エネルギーの $\gamma$ -surfaceは表現されてきた。しかし、2体間ポテンシャルで表される hcp-Ti の $\gamma$ -surface は、図 1.1 から分かる様に F 点 (stable fault) が低い位置で表されている。しかし、これは現実の hcp-Ti の変形挙動からは大きくはずれた結果であり、F 点は本来なら2体間ポテンシャルでの結果の位置よりも高い位置にある事が予測されている。

そこで本研究では、原子の種類だけから電子構造を求め、色々な物性を予測する、第一原理計算を用いて面欠陥エネルギーを計算し、 $\gamma$ -surface を再現して2体間ポテンシャルの結果と比較する [3]。

面状の欠陥に伴うエネルギー変化を表現する図として $\gamma$ -surface が知られている。これは層状に積んだ格子模型の一面を境にその上下の格子を特定の方向にずらす。そのずれに伴ってどのようなエネルギー変化をなすかをプロットした図である。これによってどのような面状のずれが起こり易いかが一目で分かる。下の図 1.1 は2体間ポテンシャルで求められた $\gamma$ -surface である。

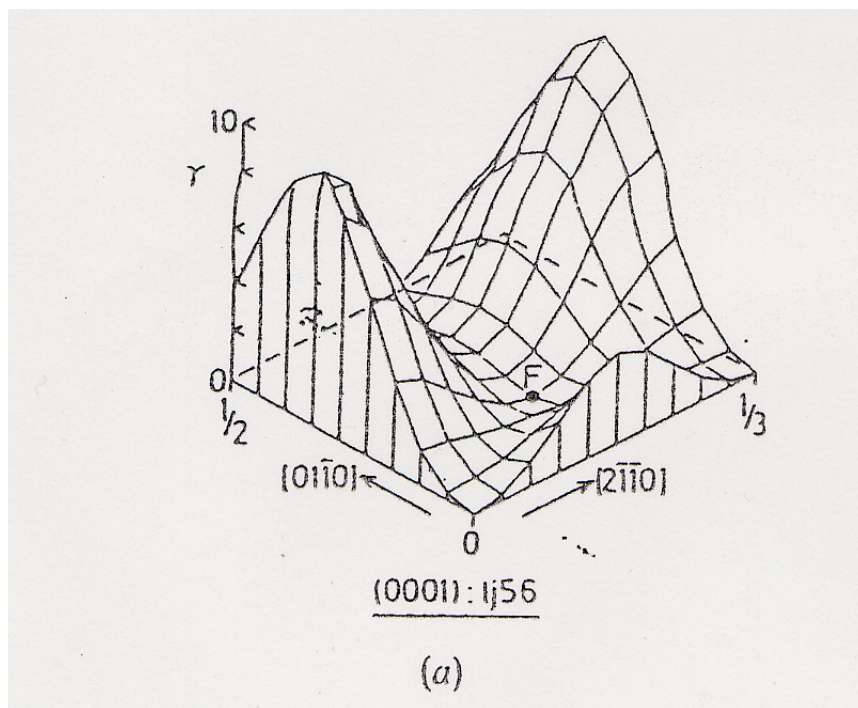


図 1.1: 2体間potential の $\gamma$ -surface

## 第2章 計算手法

### 2.1 unit cellでのTiの第一原理計算の精度の検証

何故 unit cell でポテンシャルの依存性を確かめるかというと，次の3つの理由がある [2].

- 結合エネルギーの精度を決めるため.
- 計算精度の高いFP-LMTO法と比較するため.
- 計算条件を決めるため.

#### 2.1.1 それぞれの立体構造

Tiは温度・圧力により図2.1.1に示したように，bcc,hcp,  $\omega$  構造を取る.

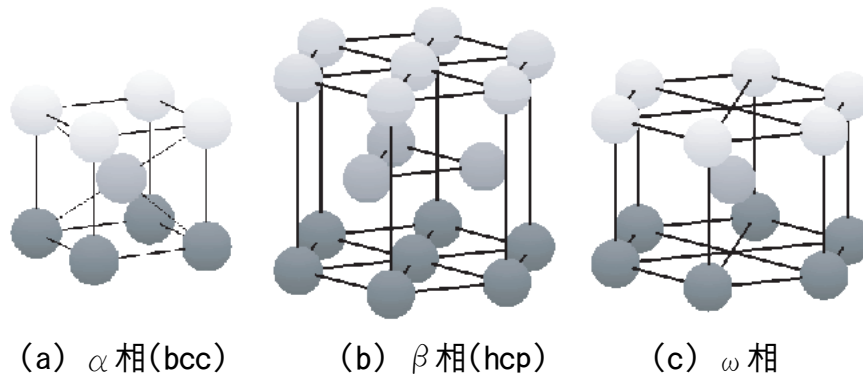


図 2.1: bcc,hcp および $\omega$  結晶の構造.

GGA,LDA,PBE,US という4種類の原子ポテンシャルでそれぞれの構造における結合エネルギーを第一原理計算で求め，計算精度を確かめる．結果，体積を横軸，エネルギーを縦軸に取ってグラフ化する．

以下に本研究で使用した bcc, hcp,  $\omega$  構造の原子位置ファイル (POSCAR) を示しておく.

bcc 構造は原子位置ファイルでは, 以下のように表される.

```
POSCAR generated by cell class
1.0
  3.2900000000000000    0.0000000000000000    0.0000000000000000
  0.0000000000000000    3.2900000000000000    0.0000000000000000
  0.0000000000000000    0.0000000000000000    3.2900000000000000
2
Direct
  0.0000000000000000    0.0000000000000000    0.0000000000000000
  0.5000000000000000    0.5000000000000000    0.5000000000000000
```

前項の原子位置ファイル内の「1.0」は, ユニットセルの倍率を表している. ユニットセルを拡大したい時は「1.05」, 縮小したい時は「0.95」という様に変換すると, ユニットセルを変化させる事ができる. 次に

```
  3.2900000000000000    0.0000000000000000    0.0000000000000000
```

という数字はそれぞれ左から x 軸, y 軸, z 軸を表している. つまり  $(x, y, z) = (3.29, 0.0, 0.0)$  という意味である. 次に, Direct の上の「2」は原子の個数を表している. そして, Direct の下の数字は原子の位置を示している.

以上からこの bcc 構造は 1 辺が 3.29(a.u) の立方体で原子が  $(x,y,z) = (0,0,0), (0.5,0.5,0.5)$  の 2 つの位置にあるユニットセルを表している.

bcc 構造と同様に hcp 構造の原子位置ファイルを紹介する.

```
POSCAR generated by cell class
1.0
  2.5547750000000000   -1.4750000000000000    0.0000000000000000
  0.0000000000000000    2.9500000000000000    0.0000000000000000
  0.0000000000000000    0.0000000000000000    4.6800000000000000
2
Direct
  0.3333000000000000    0.6667000000000000    0.2500000000000000
  0.6667000000000000    0.3333000000000000    0.7500000000000000
```

同様に  $\omega$  構造の原子位置ファイルを紹介する.

```
POSCAR generated by cell class
1.0
  4.0096980000000000   -2.3150000000000000    0.0000000000000000
```



0.0000000000000000	4.6300000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000	2.8100000000000000
1		
Direct		
0.0000000000000000	0.0000000000000000	0.0000000000000000
0.3333000000000000	0.6667000000000000	0.5000000000000000
0.6667000000000000	0.3333000000000000	0.5000000000000000

## 2.1.2 計算精度の高いFP-LMTO法

### FP-LMTO法とは

LMTO(Linear Muffin-Tin Orbital)法は、マフィンティン(MT:Muffin-Tin)球と呼ばれる原子核を中心とした球を考え、その中でシュレディンガー方程式を数値計算し、MT球の外側はポテンシャルによる影響が殆ど無いのものとして、ポテンシャルを一定(計算の簡略化の為、MT球の外側をエネルギーの原点にする事が多い)と仮定して計算する方法である。

MT球の体積は、ウィグナーザイツセルと同体積の球を考え、その球と同体積かそれ以下(MT球同士は重ならないようにする)になる。その際、MT球のポテンシャルを球対象に近似する原子球近似法が行われる。また、原子のポテンシャルが異方性を持っていて、球対称に近似してしまうと計算精度が悪くなる場合には、MT球のポテンシャルを球対称に近似せず、ポテンシャルの形状を保存して計算を行う方法もある。この場合の計算方法をフルポテンシャル(FP:Full-Potential)法という。

## FP-LMTO 法によって得られた結果

次の図は FP-LMTO 法によって計算した Ti の bcc, hcp,  $\omega$  構造それぞれの体積-エネルギー依存性である。特徴としては以下の 3 点が読み取れる。

- hcp 構造が最安定。
- bcc 構造は常に hcp 構造よりエネルギーが高い。
- 圧力を上げる（体積を下げる）と  $\omega$  構造が最安定。

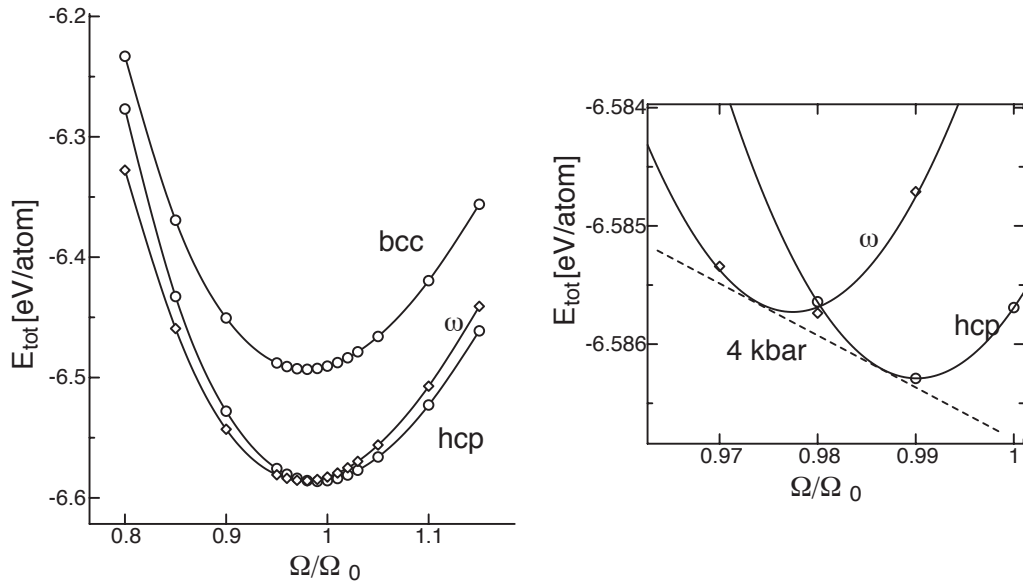


図 2.2: Ti の bcc, hcp,  $\omega$  相の結合エネルギーの体積依存性の第一原理計算。

右の図は、hcp 構造の周りを拡大した図である。hcp 相と  $\omega$  相の共存する圧力は図の共通接線の傾きから、 $p = -dE/dV$  で求まる。

### 2.1.3 VASP での計算結果

LDA とは Local Density Approximation の略で，電子状態計算で用いられる局所密度近似の事である．この LDA ファイルを使って計算した結果が図 2.3 で表される．図 2.3 と FP-LMTO 法の結果を比較すると，まず  $\omega$  構造が転移してしまっている点が大きく違う．その他は，GGA ファイルと同じ特徴が見られる．

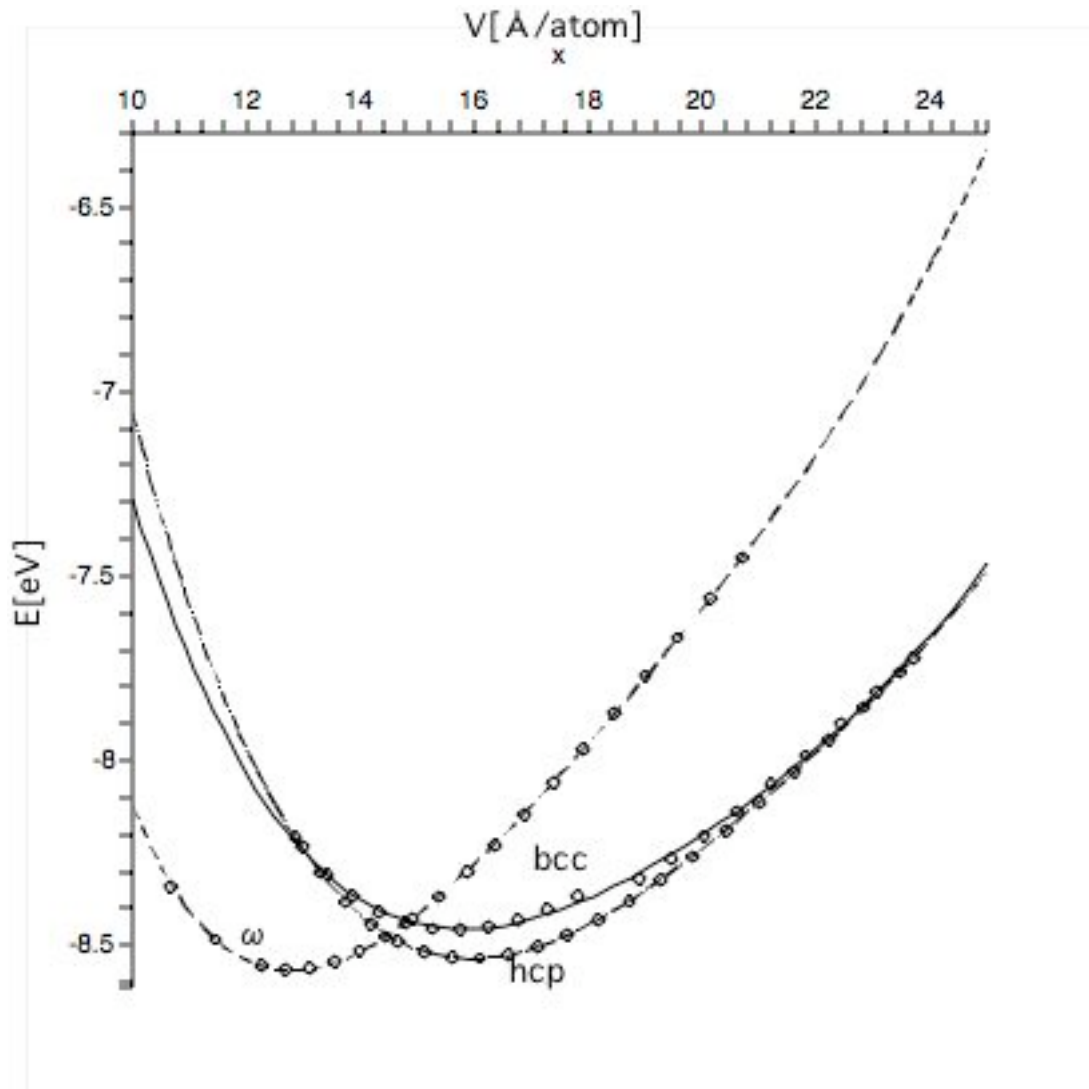


図 2.3: 原子ポテンシャル：LDA の Ti のエネルギー曲線.

GGA とは Generalized gradient approximation の略で，電子状態計算で用いられる局所密度近似 (LDA) を超える試みの一つとされている．この GGA ファイルを使って計算した結果が図 2.4 で表される．図 2.4 と FP-LMTO 法の結果を比較すると，1.  $\omega$  構造が最安定となっている．2. 体積を下げると hcp 構造が bcc 構造よりもエネルギーが高くなる．以上の 2 点が 2.1.2 で示した FP-LMTO 法で得られた特徴と違う点である事が分かる．

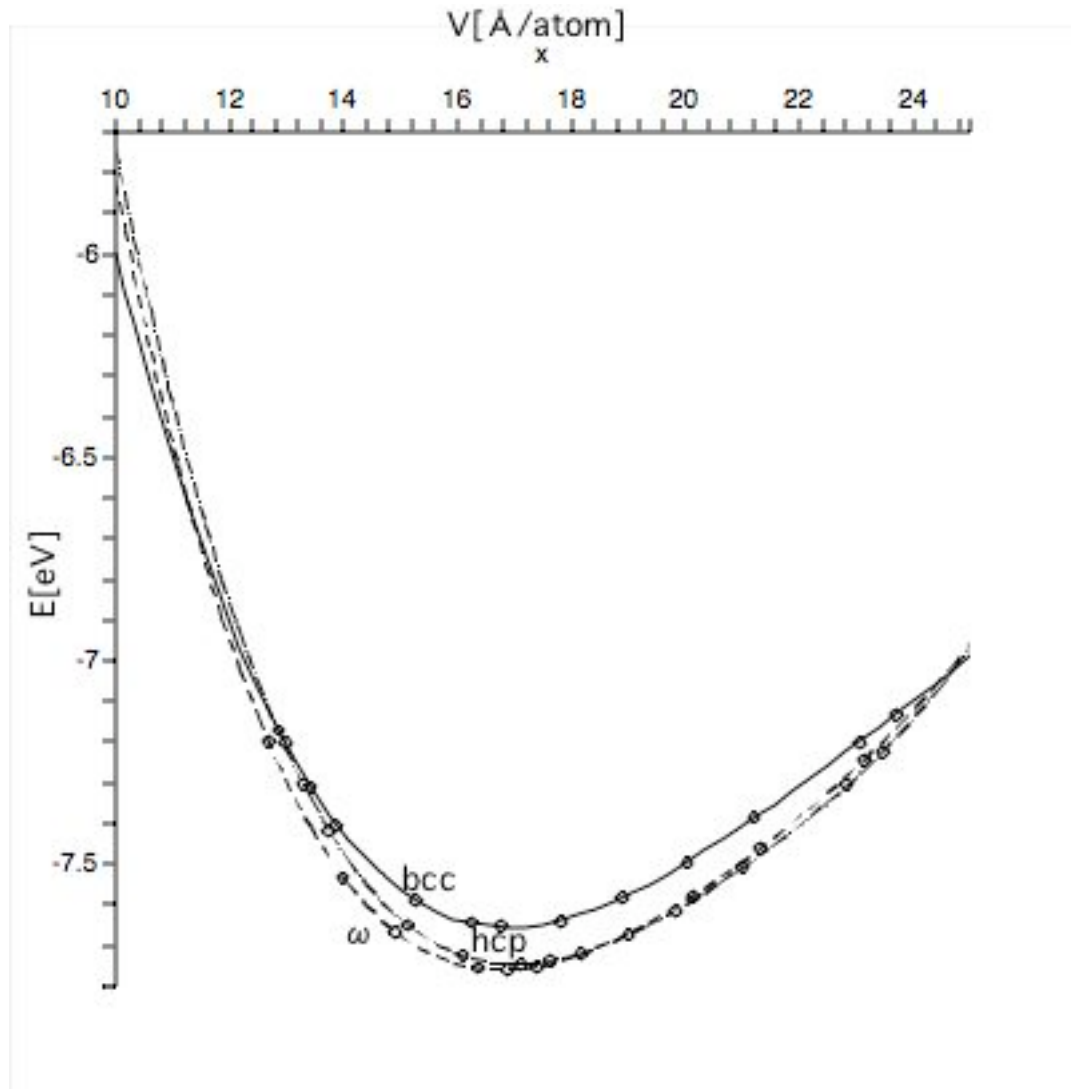


図 2.4: 原子ポテンシャル：GGA の Ti のエネルギー曲線.

PBE は、GGA の派生版である。この PBE ファイルを使って計算した結果が図 2.5 で表される。図 2.5 と FP-LMTO 法の結果を比較すると、LDA ファイルと同様に  $\omega$  構造が転位してしまっている点が大きく違う。また、LDA ファイルでの結果と似ているが、bcc 構造が hcp 構造よりも常にエネルギーが高いという特徴は、2.1.2 で示した特徴と合致する。

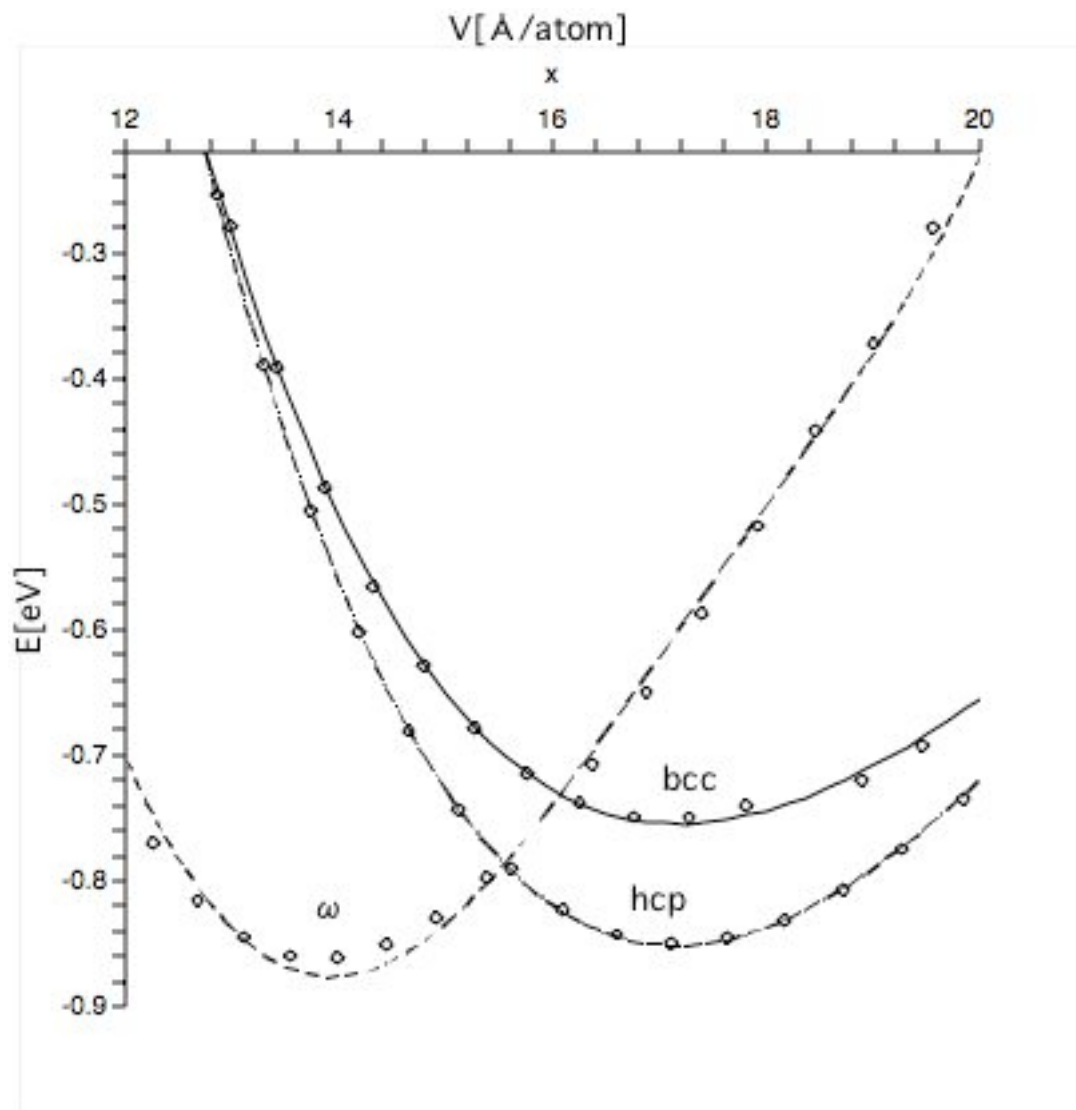


図 2.5: 原子ポテンシャル：PBE の Ti のエネルギー曲線.

US は, Ultra Soft の略である. この US ファイルを使って計算した結果が図 2.6 で表される. 図 2.6 と FP-LMTO 法の結果を比較すると, LDA,PBE ファイルと同様に  $\omega$  構造が転位してしまっている点が大きく違う. また, 体積を大きくしたり, 小さくしたりすると, bcc 構造が hcp 構造よりもエネルギーが低くなってしまう特徴がみられる.

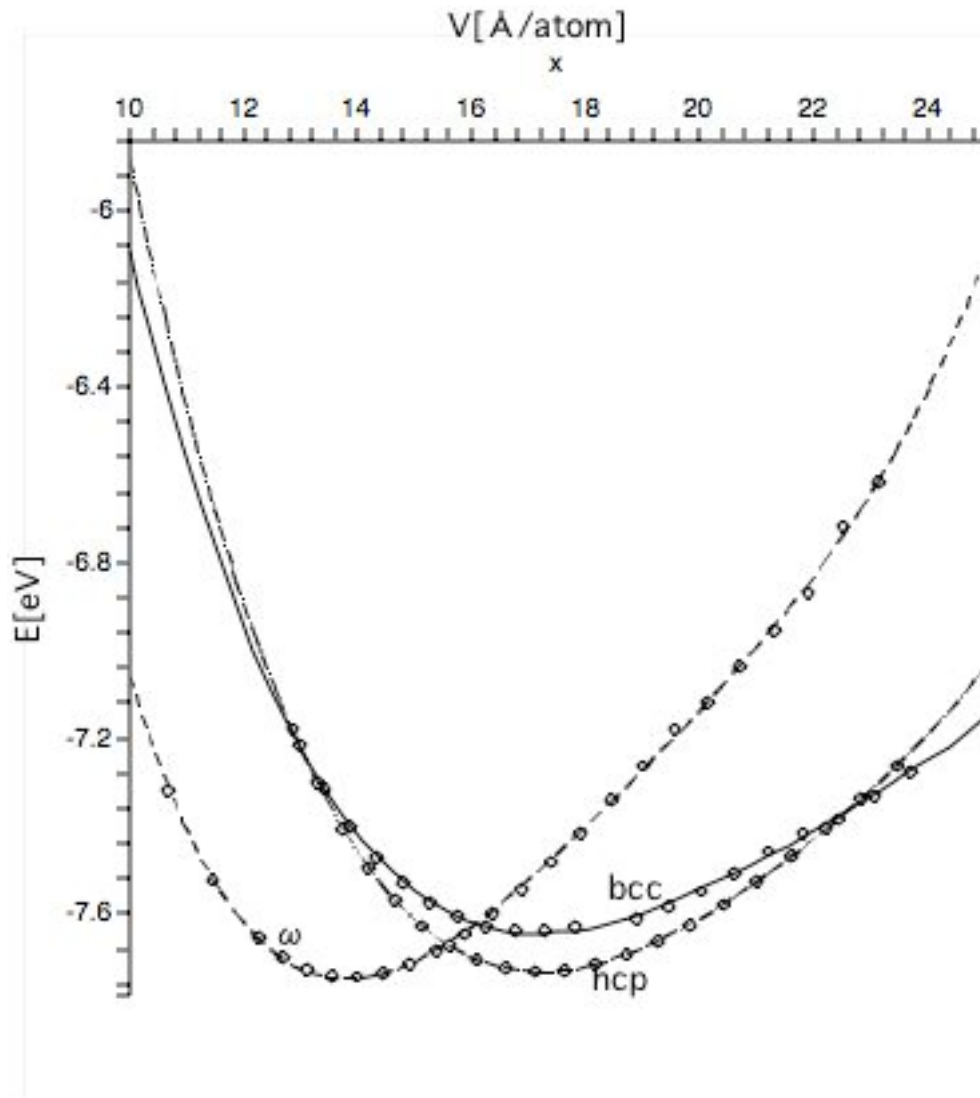


図 2.6: 原子ポテンシャル: US の Ti のエネルギー曲線.

## 2.1.4 まとめ

前頁に示した4つのグラフとFP-LMTO法の結果を比較して、どの原子ポテンシャルを使って今後の研究を進めていくかを決定する。

前述した様に、FP-LMTO法の特徴である

- hcp 構造が最安定.
- bcc 構造は常に hcp 構造よりエネルギーが高い.
- 圧力を上げる（体積を下げる）と  $\omega$  構造が最安定.

の3つを全て満たす原子ポテンシャルはなかった。

しかし、このままでは研究が進まないなので、最も近いものを選び今後の研究ではその原子ポテンシャルを使う事にする。比較するとGGAポテンシャルが一番近い事が分かる。GGAポテンシャルは $\omega$ 構造が最安定となっているが、hcp構造も殆ど $\omega$ 構造と同じぐらいエネルギーが低い。次に、体積を小さくした時にhcp構造よりもbcc構造が安定化するが、許容の範囲と言える。また、体積を下げると $\omega$ 構造が最安定となっているので、この条件は満たしている。

以上の考察より、以後の研究はGGAポテンシャルを使う事に決定した。GGAポテンシャル以外の原子ポテンシャルは、 $\omega$ 構造が転移してしまっているので、使えない事が分かった。

## 2.2 縦に積んだ時の計算精度の検証

何故縦に積んで結合エネルギーの計算精度を検証するかと言うと、面欠陥エネルギーを計算する時に必要であるからである。第一原理計算では、周期的境界条件が必須である。面欠陥エネルギーあるいは、その構造を現実的に再現するには周期的境界条件を課した時に2枚の面同士の相互作用が働かない様に十分に距離を取る必要がある。従って原子構造モデルでよく採用されるスラブモデルを用いる。このスラブモデルを周期的境界条件の下で再現するには、unit cell を縦に積んで構造を造る必要がある。

### 2.2.1 unit cell を縦に積む方法

2章の初めで示した原子位置ファイルを書き換える事でunit cellを増やす事が出来る。

また、常温常圧ではTiはhcp構造を取るため今後はhcp構造を使って研究を進める。以下に、hcp構造のunit cellを4つ積んだ時の原子位置ファイルを例を示す。

例：unit cell 4つの時の原子位置ファイル

Tihcp

1.0

```
2.55477500000 -1.475000000 0.000000000
0.000000000 2.950000000 0.000000000
0.000000000 0.000000000 18.720000000
```

8

Direct

```
0.33333 0.66667 0.06250
0.66667 0.33333 0.18750
0.33333 0.66667 0.31250
0.66667 0.33333 0.43750
0.33333 0.66667 0.56250
0.66667 0.33333 0.68750
0.33333 0.66667 0.81250
0.66667 0.33333 0.93750
```

unit cell 1つの時とどこを変えるかというと、体積を表す値のz軸の数値が(0,0,4.68)から(0,0,18.72)に変える。これは、4つのunit cellが積まれたので $4.68 \times 4 = 18.72$ となっている。次に、原子の位置を表す数値を変える。1つのunit cell内に2つの原子があるので、4つのunit cell内には8つの原子が存在するのでDirectの上の原子数を表す数字を2から8に変える。最後に原子のz軸の位置を変えなければならない。ちなみに、z軸の値は1以内で表さなければならない。以下に、分かりやすい様に図とどれだけunit cellを増やしても簡単に変換できる様にプログラムを示しておく。

```
> restart;
```

```
> N:=8; #原子数
```

```
N := 8
```

```
> for i from 1 by 2 to N do
```

```
>   x0:=[1/3,2/3,(2*i-1)/(N*2)]:
```

```
>   x1:=[2/3,1/3,(2*i+1)/(N*2)]:
```

```
>   printf("%10.5f %10.5f %10.5fn",x0[1],x0[2],x0[3]);
```

```
>   printf("%10.5f %10.5f %10.5fn",x1[1],x1[2],x1[3]);
```

```
> end do:
```

```
0.33333 0.66667 0.06250
0.66667 0.33333 0.18750
0.33333 0.66667 0.31250
0.66667 0.33333 0.43750
0.33333 0.66667 0.56250
```



0.66667	0.33333	0.68750
0.33333	0.66667	0.81250
0.66667	0.33333	0.93750

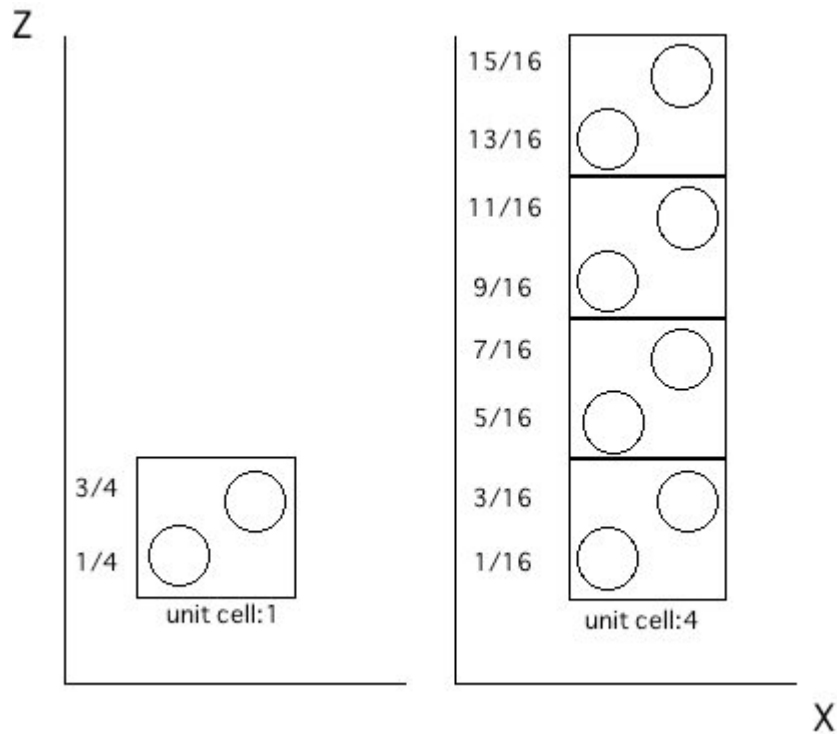


図 2.7: unit cell を横から見た図

図2.7の左図に書かれている  $1/4, 3/4$  は unit cell 1 つの時の原子の  $z$  軸の値である。右図には unit cell を 4 つ積んだ時の様子と、その時の原子の  $z$  軸の値 ( $1/16, 3/16, \dots, 15/16$ ) を示した。unit cell を増やした時に原子の  $z$  軸の値をどの様に変化させるかという、以下の式で簡単にできる。

$a := (2*i-1)/(N*4)$  #  $N$  は unit cell の数  
 $b := (2*i+1)/(N*4)$  #  $i$  は 1 から  $2N$  まで 2 ずつ変化させる

## 2.2.2 unit cellが増えた時の計算時間と精度

### 計算機の数

体積が増加すると計算時間が増加してしまう為、研究に時間がかかってしまう。そこで、計算時間を減らす方法として 2 つある。1 つは、計算機の数を増やして並列で計算する。計算機数に関しては、1, 2, 4, 8 という様に  $2^n$  数でしか計算できない。計算機を増やせば増やすほど計算時間が短くなると思われるかもしれないが、計算機を 8 つにすると、計算速度が飽和してしまうので、最適な計算機の数 は 4 つである事が分かった。

## 計算精度

計算時間を短くするもう1つの方法は、KPOINTS の mesh size を減らす事です。ここで、KPOINTS について説明します。

### KPOINT File

#### Automatic mesh

0		KPOINT の数
Monkhorst Pack		Monkhorst Pack の選択
8 8 8		mesh のサイズ
0. 0. 0.		k-mesh のシフト

1 行目の KPOINT の数は、より小さい数か 0 で与えなければならない。一般的には 0 にする。理由は、0 にしておくとベクトルの方向を自動的に生成してくれるためである。

2 行目の Monkhorst Pack は mesh のタイプを入力するところ。Monkhorst の他にも generating Monkhorst Pack などがある。

3 行目はメッシュのサイズを入力するところ。unit cell のそれぞれの方向 (x,y,z 軸) において、分割する数。

4 行目は k-mesh のシフトを入力するところ。mesh タイプが Monkhorst であれば、0 を入力。

unit cell を縦方向 (正の z 軸方向) に積む時は、左 2 つの数字はワンセットで変化させるようにする。例えば (8,4,4) などと表記しない。mesh-size を小さくすると、計算時間は短くなるが計算精度が悪くなってしまう。逆に、mesh-size を大きくすると、計算時間は遅くなるが計算精度はよくなる。計算時間がかかっても、精度が良い方を選択すべきかもしれないが、1 点を計算するのに何十時間も掛けていては研究が終わらない。そこで、なるべく計算時間を短くし、なおかつ計算精度を許容できる範囲での誤差におさめられるような mesh-size を選択する必要がある。そこで次に、具体的に unit cell を増やして計算精度と計算時間について検証した結果をまとめる。

### 2.2.3 unit cell を縦に積んだ時の計算精度の結果

#### unit cell:2

計算時間を纏まとめたものは後ほど表記するので，ここでは KPOINTS の mesh-size によってどれほど計算結果が変わるのかを示す．＜考察＞

表 2.1: 様々な k-mesh:2unit cell

k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$	k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$
1 1 1	15.12	-8.79	8 8 2	15.12	-7.65
	17.64	-9.28		17.64	-7.74
	20.42	-9.40		20.42	-7.56
2 2 2	15.12	-7.38	8 8 4	15.12	-7.65
	17.64	-7.55		17.64	-7.73
	20.42	-7.43		20.42	-7.56
4 4 4	15.12	-7.65	8 8 8	15.12	-7.65
	17.64	-7.74		17.64	-7.73
	20.42	-7.56		20.42	-7.56

k-mesh を (1 1 1) に設定すると，計算時間は短くなるが計算精度がわるくなる．実際今回の計算結果から分かる．他の k-mesh と比べると 2.25[eV] 程度の誤差がある事が分かる．次に，k-mesh を (2 2 2) 設定すると (1 1 1) の時に比べて精度はよくなったものの，他の k-mesh と比べるとまだ 0.1-0.3[eV] 程度の誤差がある．そして，(1 1 1) と (2 2 2) 以外の k-mesh では，殆ど誤差がなくなる．体積が 17.64 の時に k-mesh(4 4 4),(8 8 4) と (8 8 8),(8 8 2) の間で 0.01 の誤差があるが，これは許容の範囲と言えるのでこの 4 種類の k-mesh を使えば計算誤差については問題ないと考えられる．時間に関しては，後ほど考慮する．

unit cell:4

表 2.2: 様々な k-mesh:4unit cell

k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$	k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$
1 1 1	15.12	-8.76	8 8 2	15.12	-7.65
	17.64	-9.30		17.64	-7.74
	20.42	-9.41		20.42	-7.56
2 2 2	15.12	-7.38	8 8 4	15.12	-7.65
	17.64	-7.55		17.64	-7.74
	20.42	-7.43		20.42	-7.56
4 4 4	15.12	-7.64	8 8 8	15.12	-7.65
	17.64	-7.74		17.64	-7.73
	20.42	-7.56		20.42	-7.56
4 4 1	15.12	-7.38			
	17.64	-7.73			
	20.42	-7.64			

#### <考察>

k-mesh を (1 1 1) に設定すると, unit cell:2 の時と同様に計算精度が悪くなる. 他の k-mesh と比べても分かる様に 2 2.5[eV] 程度の誤差がある. 次に, unit cell:2 の時と同様に k-mesh を (2 2 2) 設定すると unit cell:2 の時と同様に (1 1 1) の時に比べて精度は良くなったものの, 他の k-mesh と比べるとまだ 0.1 0.3[eV] 程度の誤差がある. そして, 今回は unit cell:2 の時には試さなかった k-mesh(4 4 1) も試してみた. 体積が 17.64 の時は, 他の精度の良い結果と変わらないが, 体積が 15.12 と 20.42 の時は, 0.1 0.2 程度誤差が出ている. そして, (1 1 1),(2 2 2)(4 4 1) 以外の k-mesh を見てみると, unit cell:2 の時と同様殆ど誤差がない事が分かる. 体積が 17.64 の時に k-mesh(8 8 8) と (4 4 4)(8 8 4),(8 8 2) の間で 0.01 の誤差が見られるが, これは許容の範囲と言えるのでこの 4 種類の k-mesh を使えば計算誤差については問題ないと考えられる. また, unit cell を 4 つに増やしても計算精度に問題がない事が分かった.

unit cell:8

表 2.3: 様々な k-mesh:8unit cell

k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$	k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$
1 1 1	15.12	-8.74	8 8 1	15.12	-7.65
	17.64	-9.29		17.64	-7.74
	20.42	-9.41		20.42	-7.56
2 2 2	15.12	-7.38	8 8 2	15.12	-7.65
	17.64	-7.55		17.64	-7.74
	20.42	-7.43		20.42	-7.56
4 4 4	15.12	-7.64	8 8 4	15.12	-7.65
	17.64	-7.74		17.64	-7.74
	20.42	-7.56		20.42	-7.56
			8 8 8	15.12	-7.65
				17.64	-7.74
				20.42	-7.56

<考察>

今回は、新しく設定した k-mesh である (8,8,1) 以外は unit cell2,4 の場合と同じなので、考察は割愛する。k-mesh(8 8 1) の時の計算結果から、(4 4 4),(8 8 8),(8 8 2),(8 8 4) と同じ結果で精度が良い事が分かる。ちなみに、unit cell:4 の時に試した k-mesh(4 4 1) は既に計算精度がそれほど良くない事が分かっているので、今回は割愛した。

## グラフ

表 2.3 に示しただけではエネルギーの k-mesh 依存性がはっきり分らない．そこで視覚的に捉えやすくする為に体積  $V$  ごとに纏めて k-mesh 依存性を取る．unit cell サイズ 8 の場合を図 2.8 に示す．

多少プロット位置に揺らぎが見られる．これは，計算値で出た数値が小数点以下 6 桁で出力され，その数値をそのまま maple でプロットした為である．平衡体積 ( $V=17.64$ ) では，k-mesh を変えてもそれほど変化しない．しかし体積が大きいところ，例えば (222),(444) に見られるように差が存在する．前項の 3 つの表は，有効桁数で四捨五入をした数字を記載している．

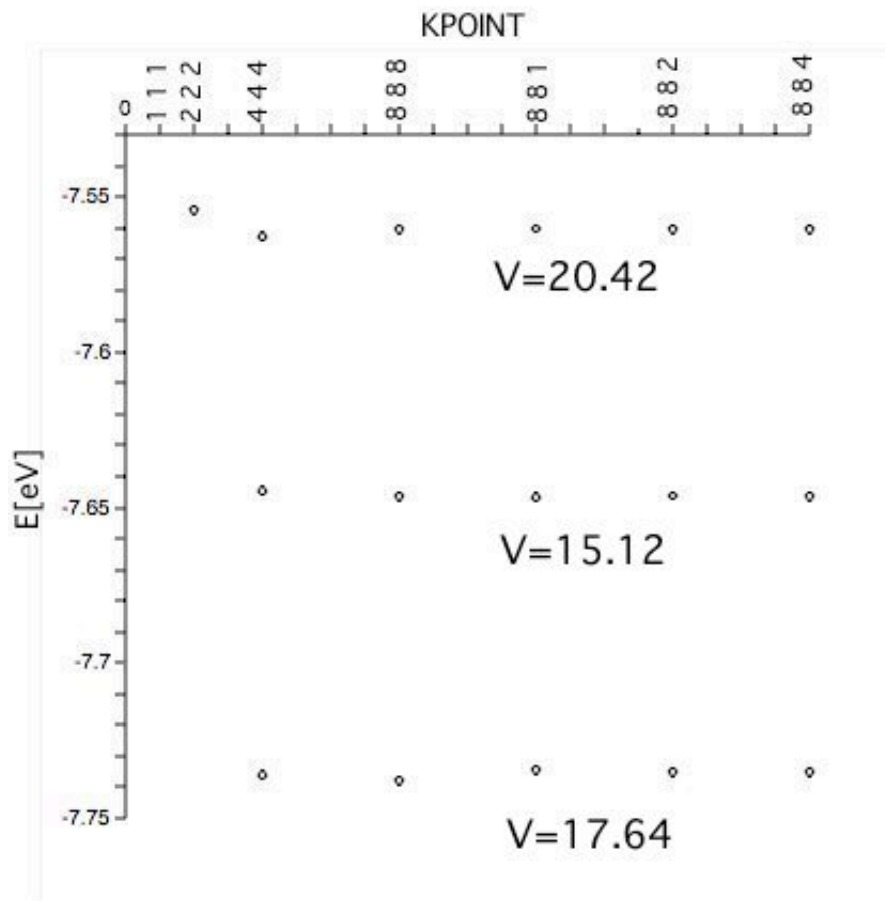


図 2.8: unit cell:8 での様々な kpoint で結合エネルギーの精度を比較.

## 2.3 まとめ

本研究では，面欠陥エネルギーは unit cell:5 で計算をした．なので，unit cell:8 で計算精度が狂っていなければそこで検証を終了しても良いが，一応 unit cell:16,32,64 においてもいくつかの k-mesh で計算精度を検証してみた．なので，以下に計算時間を含めた様々な k-mesh，unit cell の検証結果を表に纏めた．

表 2.4: 様々な k-mesh，unit cell，time

k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$	time	k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$	time
1 1 1(uc:2)	15.12	-8.79	8.2	8 8 2(uc:2)	15.12	-7.65	82.71
	17.64	-9.28	10.55		17.64	-7.73	91.62
	20.42	-9.40	28.37		20.42	-7.56	97.26
2 2 2(uc:2)	15.12	-7.38	11.08	8 8 4(uc:2)	15.12	-7.65	160.00
	17.64	-7.55	13.85		17.64	-7.74	177.14
	20.42	-7.43	16.45		20.42	-7.56	196.95
4 4 4(uc:2)	15.12	-7.65	46.71	8 8 8(uc:2)	15.12	-7.65	315.15
	17.64	-7.74	51.95		17.64	-7.73	346.98
	20.42	-7.56	55.16		20.42	-7.56	385.05
1 1 1(uc:4)	15.12	-8.76	51.89	8 8 2(uc:4)	15.12	-7.65	408.13
	17.64	-9.30	63.99		17.64	-7.74	462.64
	20.42	-9.41	67.88		20.42	-7.56	530.29
2 2 2(uc:4)	15.12	-7.38	45.84	8 8 4(uc:4)	15.12	-7.65	805.29
	17.64	-7.55	51.51		17.64	-7.74	885.03
	20.42	-7.43	57.37		20.42	-7.56	1043.8
4 4 1(uc:4)	15.12	-7.38	168.23	8 8 8(uc:4)	15.12	-7.65	1596.97
	17.64	-7.73	180.42		17.64	-7.73	1748.09
	20.42	-7.64	209.71		20.42	-7.56	2061.85
4 4 4(uc:4)	15.12	-7.64	233.19				
	17.64	-7.74	257.53				
	20.42	-7.56	290.91				



表 2.5: 様々な k-mesh, unit cell, time(続き)

k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$	time	k-mesh	$V[\text{\AA}^3]$	$E[\text{eV/atom}]$	time
1 1 1(uc:8)	15.12	-8.74	158.82	8 8 1(uc:8)	15.12	-7.65	168.65
	17.64	-9.29	199.36		17.64	-7.74	2034.49
	20.42	-9.41	217.92		20.42	-7.56	2246.92
2 2 2(uc:8)	15.12	-7.38	206.57	8 8 2(uc:8)	15.12	-7.65	1666.74
	17.64	-7.55	254.35		17.64	-7.74	2003.15
	20.42	-7.43	277.70		20.42	-7.56	2333.45
4 4 4(uc:8)	15.12	-7.64	1316.57	8 8 4(uc:8)	15.12	-7.65	3299.53
	17.64	-7.74	1510.15		17.64	-7.74	3960.170
	20.42	-7.56	1753.64		20.42	-7.56	4491.53
4 4 1(uc:8)	15.12	-7.50	1276.84	8 8 8(uc:8)	15.12	-7.65	9080.62
	17.64	-7.65	1395.35		17.64	-7.74	10967.44
	20.42	-7.32	1507.24		20.42	-7.56	10860.120
8 8 1(uc:16)	15.12	-7.64	5353.58	8 8 1(uc:64)	15.12	-7.63	160089.06
	17.64	-7.64	10256.04		17.64	-7.72	213743.77
	20.42	-7.48	9463.05		20.42	impossible	NG
8 8 2(uc:32)	15.12	-7.64	58764.08	8 8 2(uc:64)	15.12	-7.65	91632.55
	17.64	-7.73	69984.68		17.64	-7.74	49954.56
	20.42	-7.55	89852.65		20.42	-7.56	58699.65

以上のまとめた表からも分かる様に, unit cellが増えても k-mesh を (8 8 2) にすれば, 計算精度は良いと言う事が分かった.

また, unit cell:64,k-mesh(8 8 1), 体積:20.42 においても計算が強制終了されている. 以上より unit cell が多くなると, z 軸方向の mesh は 1 では不具合が生じるという予想が出来る.

本研究では予備的に計算手法を確定する事に主眼を置いているので, 計算精度と計算時間の兼ね合いから 5unit cell, k-mesh(441) で以降の計算を実行した. よりエネルギーの精度を上げるには, 表 2.5 を参照して計算パラメータを決定する必要がある.

## 第3章 結果及び考察

### 3.1 面欠陥エネルギーとは

面欠陥エネルギーとは，変形する事により金属の結晶の一部がズレた時の結合エネルギーの事である．よって，原子位置ファイル (POSCAR) で結晶の一部をずらし，その時の結合エネルギーを計算すればよい．そこで本研究では，unit cell:5 でx軸，y軸方向にそれぞれ0(ずらしていない状態)～1.0まで0.1ずつずらし，計121点の面欠陥エネルギーを計算した．

#### 3.1.1 結晶の一部をずらす方法

例:unit cell:5

以下に結晶を横にずらした模式的な図と真ん中の5原子をx,y軸方向にそれぞれ0.5ずつずらした原子位置ファイルを示す．

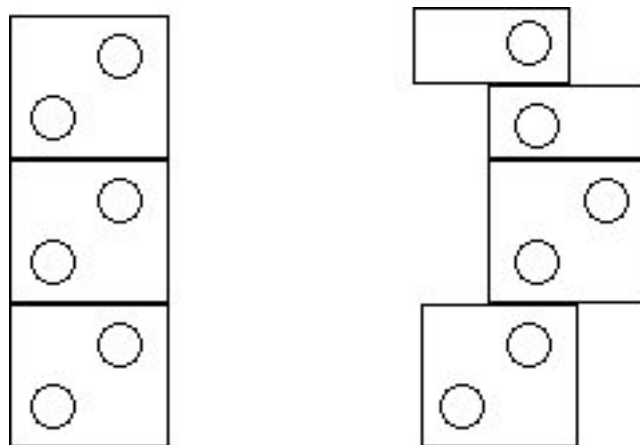


図 3.1: 左図:結晶を横から見た図. 右図:結晶の真ん中の原子をずらした図

原子位置ファイル (unit cell:5)

```
1.0
  2.5547750000000000    -1.4750000000000000    0.0000000000000000
  0.0000000000000000    2.9500000000000000    0.0000000000000000
  0.0000000000000000    0.0000000000000000    23.4000000000000000
10
Direct
  0.33333    0.66667    0.05000
  0.66667    0.33333    0.15000
  0.83333    1.16667    0.25000
  1.16667    0.83333    0.35000
  0.83333    1.16667    0.45000
  1.16667    0.83333    0.55000
  0.83333    1.16667    0.65000
  0.66667    0.33333    0.75000
  0.33333    0.66667    0.85000
  0.66667    0.33333    0.95000
```

以下にこのずらした原子の位置を簡単に求められるプログラムを示す.

```
> restart;
> N:=10:#原子の数 (unit cell の数ではない)
> #x0:=[0.33333,0.66667,1/(N*2)];
> #x1:=[0.66667,0.33333,3/(N*2)];
> DisplaceSlab1:=proc(dx1,dy1,N1,N2)
>
> local i,j,x0,x1;
> global N;
> for i from N1 by 2 to N2 do
>   x0:=[1/3,2/3,(2*i-1)/(N*2)];
>   x1:=[2/3,1/3,(2*i+1)/(N*2)];
>   printf("%10.5f %10.5f %10.5f F F Tn",x0[1]+dx1,x0[2]+dy1,x0[3]):
>   if(i=7) then
>     printf("%10.5f %10.5f %10.5f F F Tn",x1[1],x1[2],x1[3]):
>   else
>     printf("%10.5f %10.5f %10.5f F F Tn",x1[1]+dx1,x1[2]+dy1,x1[3]):
>   end if;
> end do:
> end proc:
>
```

```
> dx:=0.5:
> dy:=0.5:
> DisplaceSlab1(0,0,1,N/5):
> DisplaceSlab1(dx,dy,N/5+1,N*3/5+1):
> DisplaceSlab1(0,0,N*3/5+3,N):
```

以上の様に原子位置ファイルを変更する事だけでは、面欠陥エネルギーがきちんと計算できない2つの問題がある。以下で、その問題点について詳しく説明する。

1つ目は、図(3.1)を見て分かる様に結晶の一部をずらした事で、一部の原子が非常に近くなっている事が見て取れる。その為に上手く面欠陥エネルギーが計算できない。この問題を解決する為に、内部緩和という方法を取らなければならない。

2つ目の問題は結晶を縦に積み重ねる事で平行格子定数の unit cell で結合エネルギーが最も低くなる(最も安定する)とは限らない事である。つまり、内部緩和の影響で結晶の体積が少し大きくなるもしくは小さくなった所(±5~10パーセント程度)で最もエネルギーが低くなる可能性がある。この問題を解決する為に外部緩和という方法を取らなければならない。

## 3.2 構造緩和

構造緩和には、内部緩和と外部緩和がある。図3.1から分かる様に結晶の一部をずらした事で、一部の原子が非常に近くなっている事が見て取れる。その為に上手く面欠陥エネルギーが計算できない。この問題を解決する為に、内部緩和という方法を取らなければならない。

2つ目の問題は結晶を縦に積み重ねる事で平行格子定数の unit cell で結合エネルギーが最も低くなる(最も安定する)とは限らない事である。つまり、内部緩和の影響で結晶の体積が少し大きくなるもしくは小さくなった所(±5~10パーセント程度)で最もエネルギーが低くなる可能性がある。この問題を解決する為に外部緩和という方法を取らなければならない。

### 3.2.1 内部緩和

内部緩和(relaxation)とは、電子構造計算において最も計算量が要求されるルーチンである。内部緩和というのは通常、ユニットセル内の原子に働く力によって当該原子をより安定な位置、方向に動かすことによって遂行される。よって、先述した様に一部の原子が近寄りすぎているのを内部緩和する事で、解消出来る。構造最適化計算が終了したかどうかは、通常最適化すべき原子に働く力の値がどのくらい小さくなったかによって判定する。計算上、完全にエネルギーをゼロにするのは不可能(結晶構造の持つ対称性から力の働かない原子が存在する場合もある)である。しかし、実効上ある値より小さくなれば、もう原子はほとんど動かなくなる。

そして、この内部緩和を vasp で実現する方法は、原子位置ファイルを少し書き換える必要がある。その書き換えた原子位置ファイルを以下に示す。

原子位置ファイル (unit cell:5, 内部緩和)

```

1.0
  2.5547750000000000      -1.4750000000000000      0.0000000000000000
  0.0000000000000000      2.9500000000000000      0.0000000000000000
  0.0000000000000000      0.0000000000000000      23.4000000000000000
10
Selective dynamics
Direct
  0.33333  0.66667  0.05000 F F T
  0.66667  0.33333  0.15000 F F T
  0.83333  1.16667  0.25000 F F T
  1.16667  0.83333  0.35000 F F T
  0.83333  1.16667  0.45000 F F T
  1.16667  0.83333  0.55000 F F T
  0.83333  1.16667  0.65000 F F T
  0.66667  0.33333  0.75000 F F T
  0.33333  0.66667  0.85000 F F T
  0.66667  0.33333  0.95000 F F T

```

上記の様に Selective dynamics を書き加える事で、原子をそれぞれ移動できる様に設定する。そして、原子の位置の横に記述されている F や T はその原子を動かすか動かさないかを設定する為のもの。F ならば原子は動かさないで、T なら原子を動かすという意味である。本研究では z 軸方向に原子を動かす必要があるので、F F T と表記している。

### 3.2.2 外部緩和

外部緩和を考慮する為に, unit cell の体積を 95 パーセントから 105 パーセント程度で変化させてその中で何点かプロットして, 最小二乗法で曲線化をしその曲線から最小値 (最もエネルギーが安定する所) を求める. その手順を以下に示す. まず変化させた体積とその時の面欠陥エネルギーを表に纏める.

表 3.1: unit cell の体積変化とその時の面欠陥エネルギー

体積変化 $V[\text{\AA}]$	$E[\text{eV/atom}]$
18.720	-7.3485772
19.656	-7.5061758
20.405	-7.6002335
21.528	-7.6913594
23.400	-7.7312576
23.868	-7.7248763

そして次に, これらの数値を入力しプロットし, 最小二乗法で曲線化する. そして, その曲線を微分して極小値を求める. この問いの極小値が最もエネルギーの安定する所となる. 以下に外部緩和のプログラムと, その時の様子を図で示しておく.

```
> restart;
> with(Maplets[Examples]);
> file1:=GetFile();
> T:=readdata(file1,2):
Warning, the protected name LinearAlgebra has been redefined and unprotected
[Alert, Confirm, GetColor, GetEquation, GetExpression, GetFile, GetInput,
Integration, KernelOpts, LinearAlgebra, Message, Question, Selection,
ShowTable, SignQuery]

> p1:=T;
> with(stats):
> with(linalg):
> with(plots):
Warning, the protected names norm and trace have been redefined and unprotected
Warning, the name changecoords has been redefined

> d1:=pointplot(p1):
```

```

> data1:=convert(transpose(convert(p1,array)),listlist);
> fit1:=fit[leastsquare[[x,y], y=c0+c1*x+c2*x^2+c3*x^3+c4*x^4+c5*x^5,
  {c0,c1,c2,c3,c4,c5}]](data1);
> f1:=unapply(rhs(fit1),x);
> d2:=plot(f1(x),x=15..25,color=black):
> display(d1,d2):
> f2:=unapply(diff(f1(x),x),x);
> p3:=fsolve(f2(x)=0,x=18..25);
> f1(p3);

```

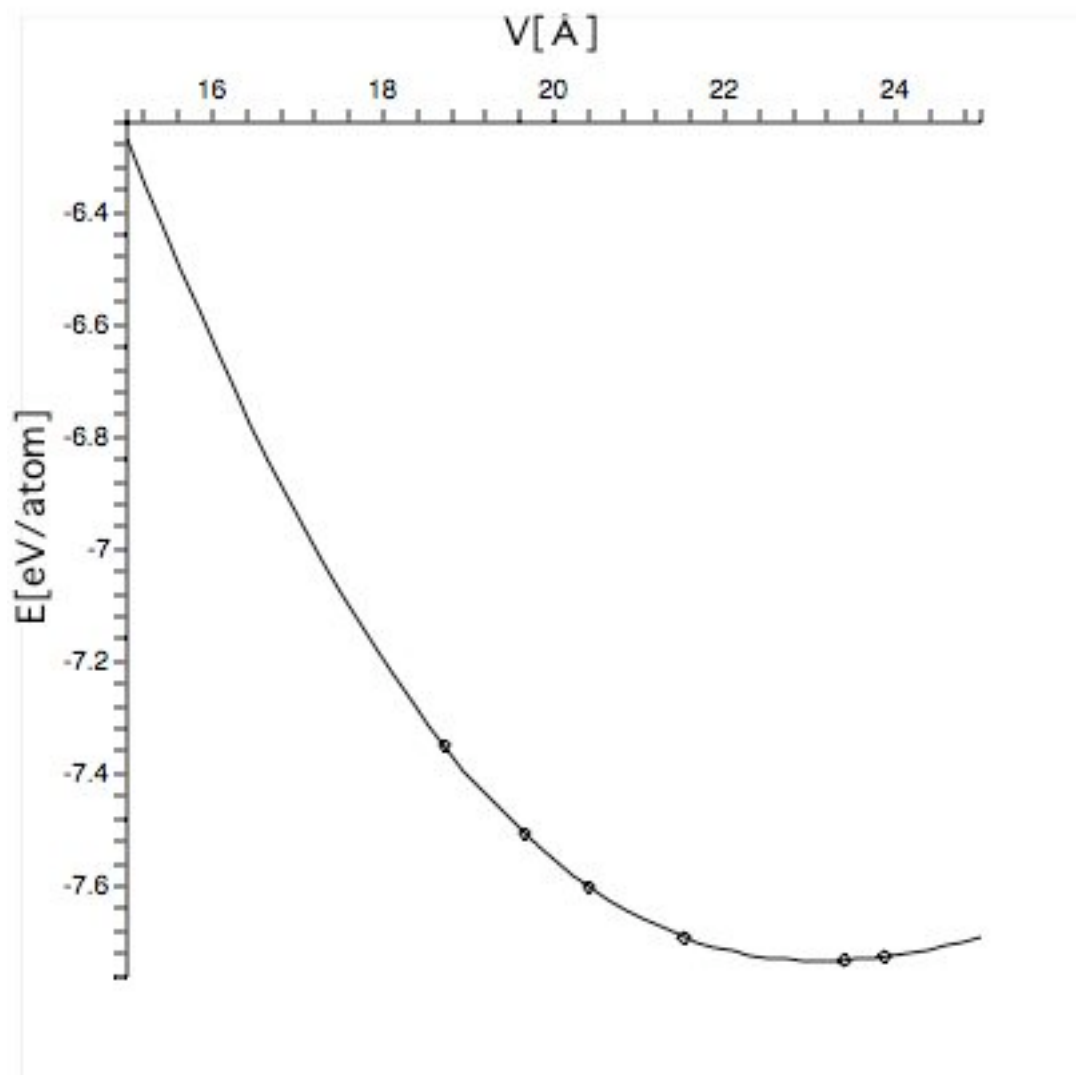


図 3.2: 外部緩和の様子

上図の例では，平衡格子定数の unit cell の体積は 23.4 でその時のエネルギーは-7.7312576 であった．しかし，外部緩和を適用すると，最も安定したエネルギーは-7.732692356 でその時の体積は 23.09354298 であった．外部緩和を適用したこの結果からは，体積で約 0.3，エネルギーで 0.0015 の差があることが分かる．



### 3.3 失敗例

前述した様に，面欠陥エネルギーを計算する時は unit cell 内の原子を奇数個ずらして計算する必要がある．最初偶数個ずらして計算したが，間違った計算結果になった．偶数個ずらした場合は，上面と下面で原子の位置が相殺されてしまい上手く面欠陥エネルギーが計算出来なかったと思われる．その失敗例の  $\gamma$ -surface と奇数個ずらした時の本来の結果を合わせてを以下に示しておく．

図 3.3 と図 3.4 は，図 3.5 の点線で囲まれた範囲で 0~1.0 まで 0.1 ずつずらして面欠陥エネルギーを計算し，縦軸をエネルギーに取り，横軸をそれぞれ  $x$ ,  $y$  軸にずらした方向にとってグラフ化したものである．前の図 3.4 では，(6,8) 辺りが一定のエネルギーになっているが，図 3.3 では (6,8) 辺りが一部エネルギーが低くなっている点がある．この低くなっている点が stable fault と呼ばれる点で，この点が現れていれば面欠陥エネルギーを計算出来ている事になる．

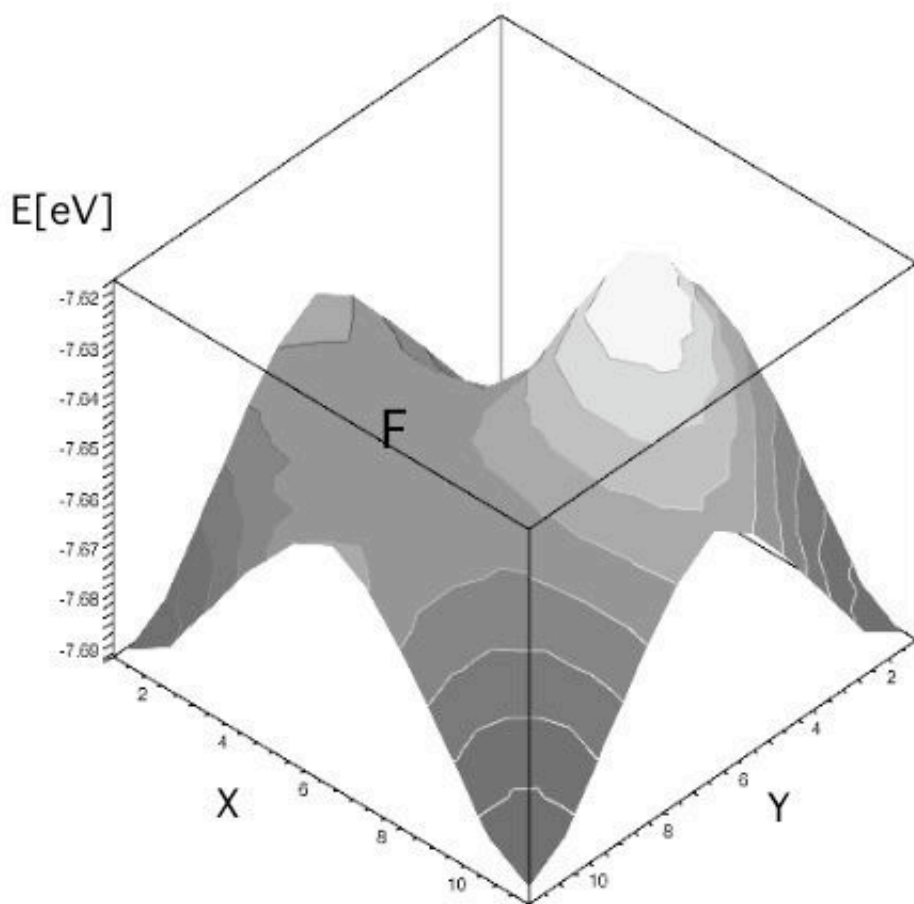


図 3.3: 失敗例

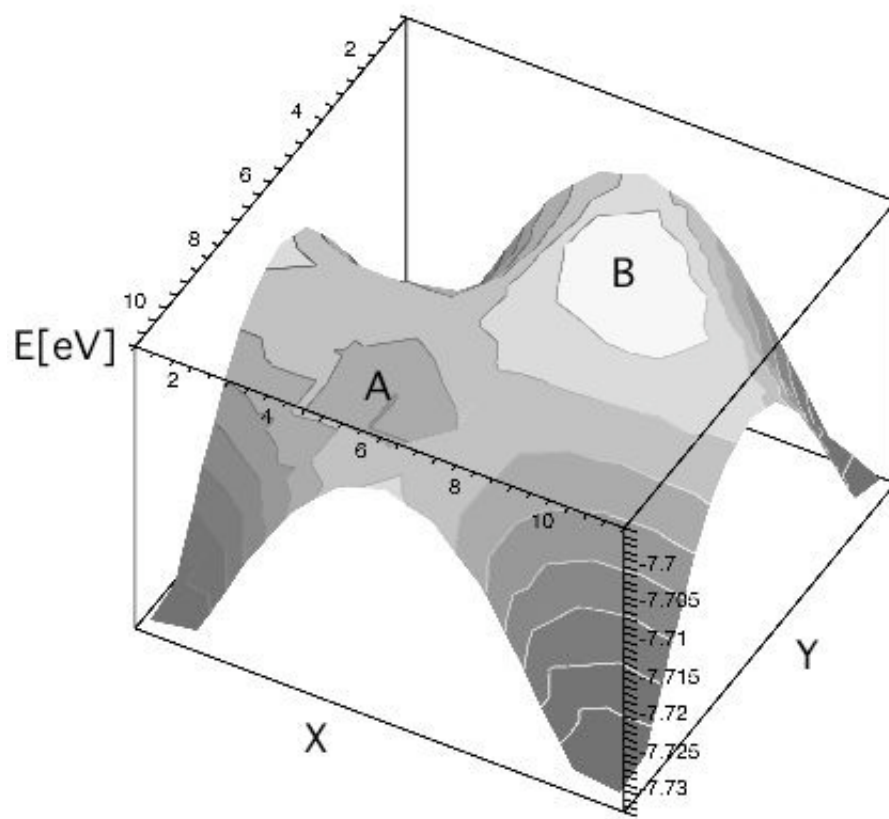


図 3.4: 成功例

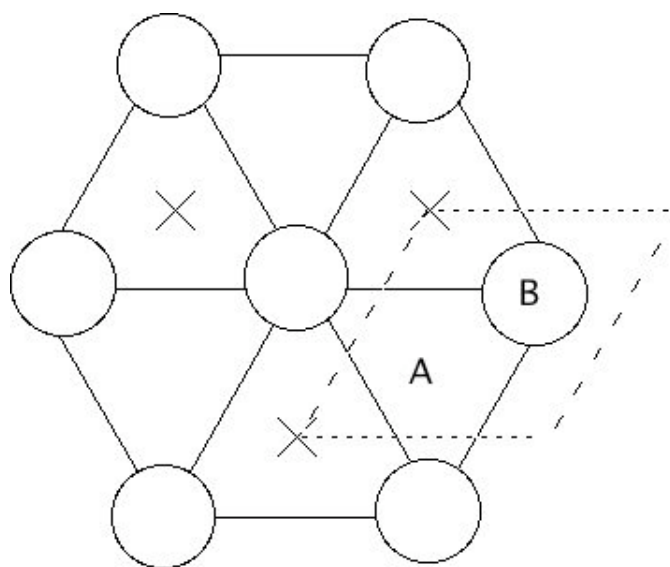


図 3.5: hcp 構造を上から見た図

### 3.4 計算結果

3.3 の図 3.3 で示した  $\gamma$ -surface は  $x, y$  方向にそれぞれ 0.1 ずつずらした面欠陥エネルギーの結果である。しかしこのままでは元来の 2 体間ポテンシャルで求められた  $\gamma$ -surface と比較出来ないで、方向を変換しなければならない。

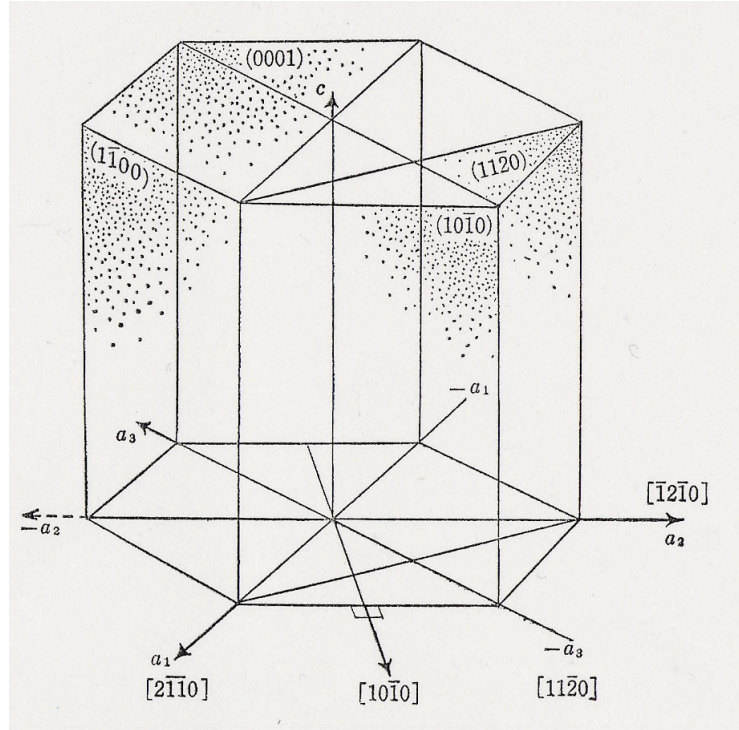


図 3.6: 境界面

本研究では、図 3.6 で示される  $(01\bar{1}0)$  面と  $(2\bar{1}\bar{1}0)$  面で表された 2 体間ポテンシャルの  $\gamma$ -surface と比較した。このままでは少し分かりにくいと思うので、もう少し分かりやすい図を 3.4.2 で示し、その変換プログラムを付録で示す。

### 3.4.1 2体間 potential での $\gamma$ -surface

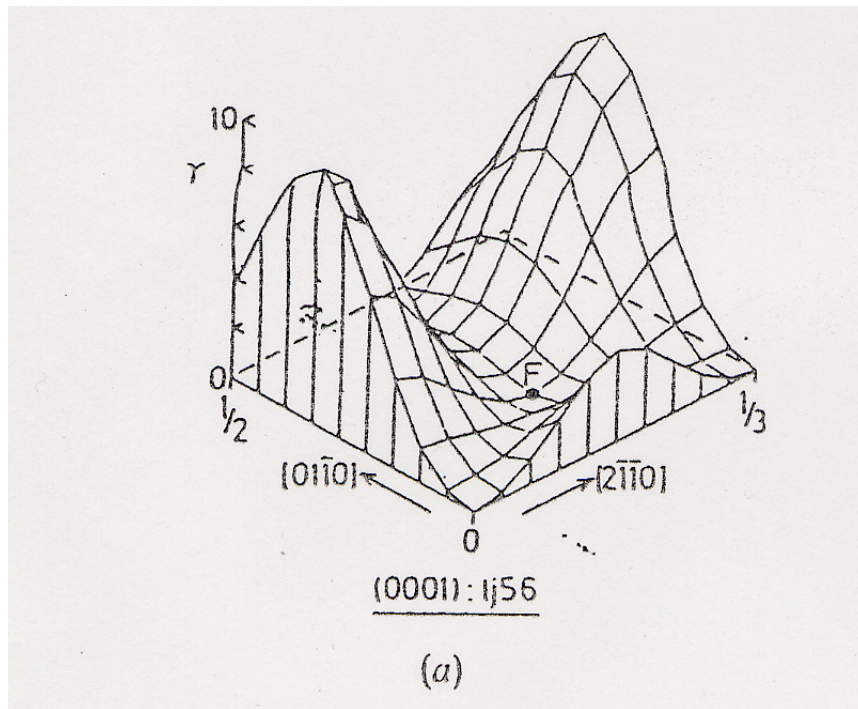


図 3.7: 2体間 potential の  $\gamma$ -surface

図を見て分かるように、2体間ポテンシャルによる  $\gamma$ -surface の特徴は、F 点 (stable fault) が非常に低い位置にあるという事である。つまり、F 点とエネルギーが最も高い点との差が非常に大きい事が分かる。そして、この結果は、本来の予測とは違い非現実的な結果となっている。

### 3.4.2 第一原理計算での面欠陥エネルギーの $\gamma$ -surface

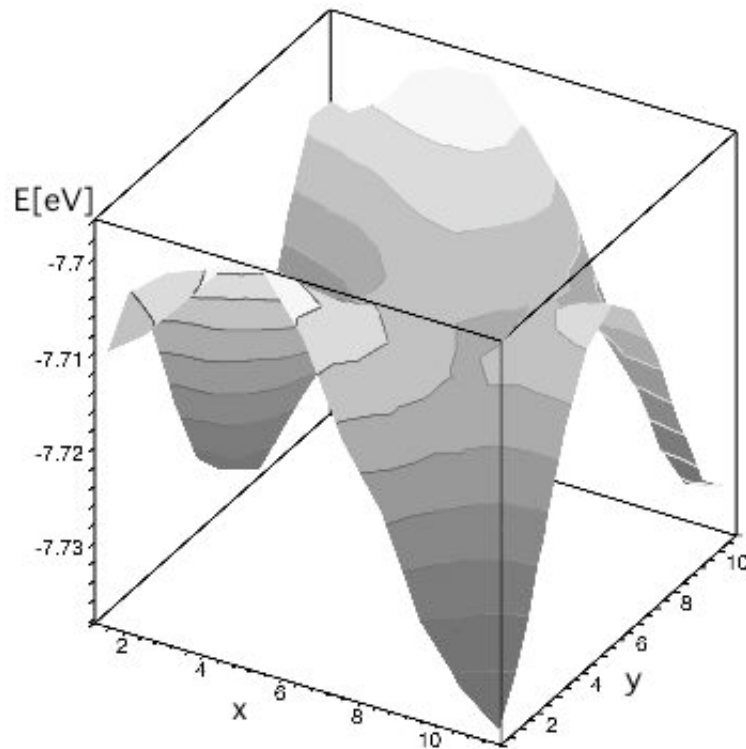


図 3.8: 第一原理計算での  $\gamma$ -surface

この図は、図 3.4 を 2 体間ポテンシャルの結果と比較できるように方向を変換し、縦軸にエネルギーを取り、横軸に  $x, y$  のずらした方向を取ってグラフ化したものである。図 3.5 から図 3.10 にどのように変化させるかという点、図 3.9 の左斜線の部分を右斜線の部分に移動させたものが図 3.10 になる。図を見ると F 点は無いように見えるかもしれないが、 $(X, Y) = (8, 4)$  辺りの周りに比べエネルギーが低くなっている所が F 点にあたる。第一原理計算の結果では、F 点が 2 体間ポテンシャルの結果に比べて高い位置にあるので、現実的な予測が再現された。

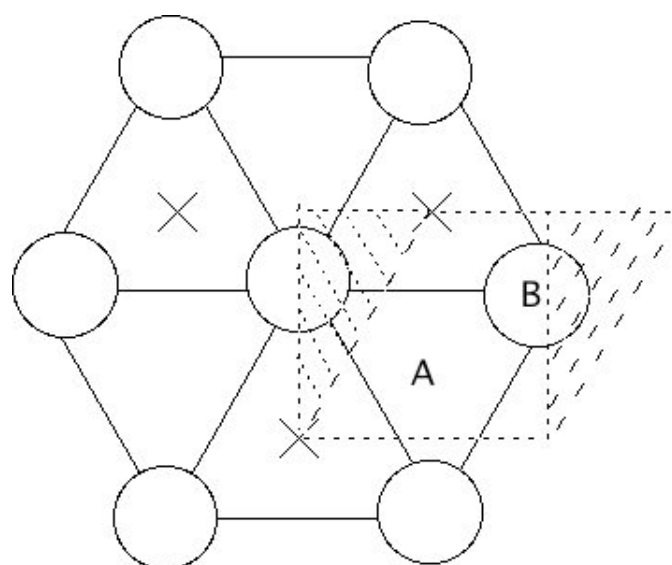


図 3.9: hcp 構造を上から見た図 2

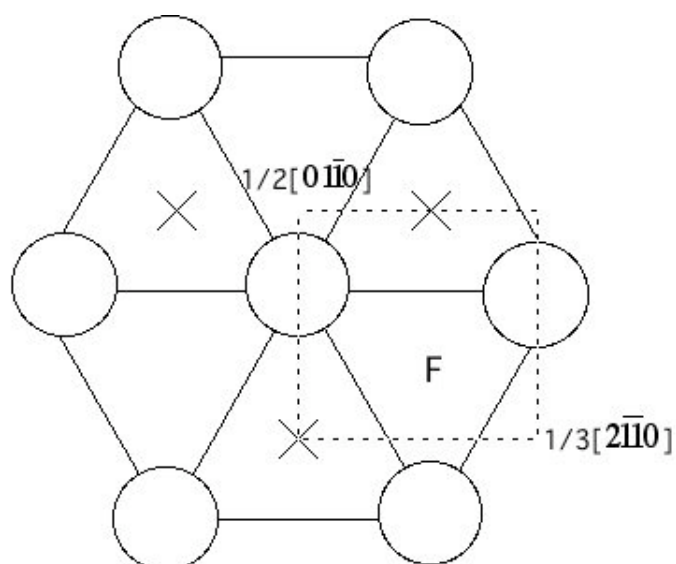


図 3.10: hcp 構造を上から見た図 3

## 第4章 まとめ

本研究で行ったことを順序立てて記述し、まとめとする。

- Ti の結合エネルギーの計算精度を検証する為に, 1unit cell で平衡格子定数の体積の $\pm 10$  パーセント程度変化させ, 第一原子計算で結合エネルギーを計算し FP-LMTO 法の結果と比較する.
- 面欠陥エネルギーの計算は, 5unit cell で行う為 unit cell を縦に積んだ時の結合エネルギーの精度を検証する.
- 面欠陥エネルギーの計算を行う. 面欠陥エネルギーを計算する際に注意しなければならない点は,
  - 奇数個の unit cell で面欠陥エネルギーを計算しなければならない.
  - 内部緩和を考慮しなければならない.
  - 外部緩和を考慮しなければならない.



図 3.7 と図 3.8 の 2 つのグラフは、同じエネルギー曲線であるが、違った手法で求められたものである。第一原理計算から求めたエネルギー曲線が今回本研究での結果である。2 体間ポテンシャルの結果とエネルギーが最も高い所と最も低い所は同じであるが、F 点 (stable fault) の位置が違う。2 体間ポテンシャルでは本来の予測とは違った結果になっていたのだが、第一原理計算の結果では F 点がもっとも高い欠陥エネルギー値の 3 分の 2 程度であり、本来の予測通りの結果が得られた。

以下に、2 体間ポテンシャルの結果である図 3.7 と第一原理計算の結果である図 3.8 を比較できるように、一つに纏めた図を示しておく。本研究においては、 $(01\bar{1}0)$  面と  $(2\bar{1}\bar{1}0)$  面で面欠陥エネルギーの計算を行ったが、他にも計算すべき面があるので、今後の研究で進めてもらいたい。

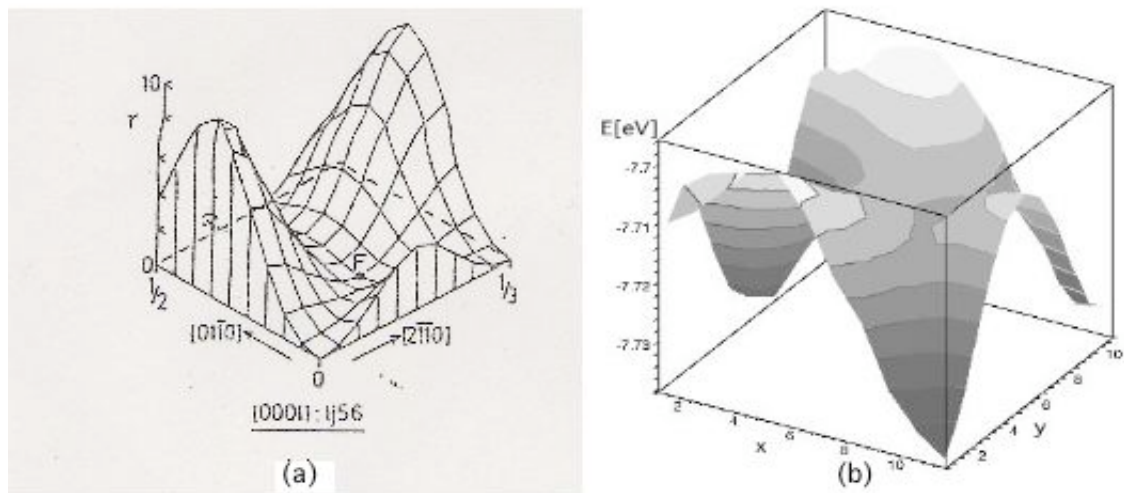


図 4.1: 2 体間ポテンシャルと第一原理計算の  $\gamma$ -surface の比較



# 付 録 A エネルギー計算の効率化

## A.1 効率化の必要性

計算精度を検証したり，外部緩和を実行する為に結合エネルギーを計算する時，unit cell の体積を±10パーセント程度変化させてその内何点かを計算した．その際，1点1点体積を表すのz軸の値を変えていたのでは，大変なので自動化するプログラムを紹介する．

## A.2 プログラム

```
#!/bin/csh -f
set Values=(          #z 軸の値を入れる.
"23.4"
"23.868")
echo "START calc" > res.all          #"START calc"に res.all を追加.

@ index=1          #1で初期化
while (${index}<3)
sed "s/#SCALE/$Values[$index]/g" POSCAR.orig > POSCAR
#time /usr/local/mpich/bin/mpirun -np 4 -machinefile machines
/usr/local/vasp/vasp4628mpi >> res.all          #timeで計算時間を表示
させ /usr/local/mpich/bin/mpirun - np 4 -machinefile machines /usr/local/vasp/
vasp4628mpiを res.all で上書きする.

time /usr/local/vasp/vasp4628serial >> res.all
echo "#BOB SCALE =" $Values[$index] >> res.all
@ index++
end
```

## A.3 原子ポテンシャルファイルの変更

POSCAR generated by cell class

1.0

2.5547750000000000	-1.4750000000000000	0.0000000000000000
0.0000000000000000	2.9500000000000000	0.0000000000000000
0.0000000000000000	0.0000000000000000	#SCALE

2

Direct

0.3333000000000000	0.6667000000000000	0.2500000000000000
0.6667000000000000	0.3333000000000000	0.7500000000000000

原子ポテンシャルファイルの z 軸の値を表す箇所に SCALE と書き, (A.2) で示したプログラムを実行する事で, set Values に書き込んだ個数分結合エネルギーが計算される.

# 付 録 B 結合エネルギー曲線の描画方法

## B.1 グラフの書き方

vasp を使って Ti の結合エネルギーの計算をする。この時、原子位置ファイルで表されている平衡格子定数の unit cell1 点のみを計算するのではなく、 $\pm 10$  パーセントの範囲で体積を変化させその内 10 点程度を計算し、縦軸を結合エネルギーに取り、横軸を体積に取ってプロットする。その後最小二乗法でグラフ化する。

## B.2 Maple でのプログラム

```
> restart;
```

```
> with(Maplets[Examples]);
> file1:=GetFile();
> A:=readdata(file1,2):
> file2:=GetFile();
> B:=readdata(file2,2):
> file3:=GetFile();
> C:=readdata(file3,2):
>
>
```

```
Warning, the protected name LinearAlgebra has been redefined and unprotected
[Alert, Confirm, GetColor, GetEquation, GetExpression, GetFile, GetInput,
Integration, KernelOpts, LinearAlgebra, Message, Question, Selection,
ShowTable, SignQuery]
```

```
file1 := "/Users/Yusuke/Desktop/Ti data/bcc vol PAW_GGA.txt"
file2 := "/Users/Yusuke/Desktop/Ti data/hcp vol PAW_GGA.txt"
file3 := "/Users/Yusuke/Desktop/Ti data/omega vol PAW_GGA. txt.txt"
> p1:=A;
> p2:=B;
```

```

> p3:=C:

> for i from 1 to nops(p1) do
> p1[i][2]:=p1[i][2]/2;
> end do:
> print(p1);
[[12.98031484, -7.196259500], [13.41781733, -7.307669500],
  [13.86504170, -7.401892000], [15.26611446, -7.588596500],
  [16.25073098, -7.642823000], [16.75853016, -7.651610000],
  [17.80564450, -7.639235000], [18.89549239, -7.581941500],
  [20.02892850, -7.494681000], [21.20680749, -7.381898000],
  [23.05882599, -7.196589500], [23.69931283, -7.127902000]]

> for i from 1 to nops(p2) do
> p2[i][2]:=p2[i][2]/2;
> end do:
> for i from 1 to nops(p3) do
> p3[i][2]:=p3[i][2]/3;
> end do:
> with(stats):
> with(linalg):
> with(plots):
Warning, the protected names norm and trace have been redefined and unprotected
Warning, the name changecoords has been redefined
> d1:=pointplot(p1,color=red):
> d2:=pointplot(p2,color=blue):
> d3:=pointplot(p3):

> display(d1,d2,d3);

> data1:=convert(transpose(convert(p1,array)),listlist);
> data2:=convert(transpose(convert(p2,array)),listlist);
> data3:=convert(transpose(convert(p3,array)),listlist);

data1 := [[12.98031484, 13.41781733, 13.86504170, 15.26611446, 16.25073098,
  16.75853016, 17.80564450, 18.89549239, 20.02892850, 21.20680749,
  23.05882599, 23.69931283], [-7.196259500, -7.307669500, -7.401892000,
  -7.588596500, -7.642823000, -7.651610000, -7.639235000, -7.581941500,
  -7.494681000, -7.381898000, -7.196589500, -7.127902000]]

```

```

data2 := [[12.85636102, 13.28968564, 13.73263930, 15.12033269, 16.09554675,
17.11181652, 17.63561182, 18.16998850, 19.83766486, 21.00429585,
22.83862874, 23.47299934], [-7.166322000, -7.301369500, -7.415715000,
-7.648524500, -7.723712500, -7.744996500, -7.738185500, -7.720874500,
-7.614901500, -7.506498000, -7.300016000, -7.222116500]]

data3 := [[12.67667186, 13.98706412, 14.90900073, 16.36650499, 16.87265024,
17.38912463, 19.00156599, 20.13008540, 21.30242540, 23.14492489],
[-7.194398000, -7.531508333, -7.666020333, -7.755532333, -7.759589000,
-7.752321000, -7.672411667, -7.578436333, -7.458100667, -7.239602000]]

>fit1:=fit[leastsquare[[x,y],
y=c0+c1*x+c2*x^2+c3*x^3+c4*x^4+c5*x^5,{c0,c1,c2,c3,c4,c5}]](data1);
>fit2:=fit[leastsquare[[x,y],
y=c0+c1*x+c2*x^2+c3*x^3+c4*x^4+c5*x^5,{c0,c1,c2,c3,c4,c5}]](data2);
>fit3:=fit[leastsquare[[x,y],
y=c0+c1*x+c2*x^2+c3*x^3+c4*x^4+c5*x^5,{c0,c1,c2,c3,c4,c5}]](data3);

> f1:=unapply(rhs(fit1),x);
> f2:=unapply(rhs(fit2),x);
> f3:=unapply(rhs(fit3),x);

> d4:=plot(f1(x),x=10..25,linestyle=1, color=black):
> d5:=plot(f2(x),x=10..25,linestyle=2, color=black):
> d6:=plot(f3(x),x=10..25,linestyle=3, color=black):
> display(d4,d5,d6);

```

## 付 録 C 境界面の変換プログラム

```
>p1:=[[a,b,..z]..[]] #計算した面欠陥エネルギーを配列 p1 に入れる.
>z:=[];
for i from 1 to 11 do
z:=[op(z),op(op(p1)[i])];
end do:

with(plots):
> listcontplot3d(p1,axes=boxed,orientation=[126,44],filled=true);

>x:=[];
y:=[];
for i from 0 to 10 do
for j from 0 to 10 do
x:=[op(x),evalf(i/10,2)];
y:=[op(y),evalf(j/10,2)];
end do;
end do;

> with(LinearAlgebra):
> p2:=convert(Transpose(Matrix([x,y,z])),listlist):
> pointplot3d(p2,symbol=circle,symbolsize=30);

>X1:=Vector([2.554775,-1.475]);
Y1:=Vector([0,2.95]);
p3:=array(p2):
n:=nops(p2);
for i from 1 to n do
v1:=convert(p2[i][1].X1+p2[i][2].Y1,list);
p3[i,1]:=v1[1];
p3[i,2]:=v1[2];
end do;
```

```

>for i from 1 to n do
  if p3[i,2]<0 then
    p3[i,2]:=p3[i,2]+2.95;
  end if;
end do;
for i from 1 to n do
  if p3[i,2]>2.95/2 then
    p3[i,2]:=p3[i,2]-2.95;
  end if;
end do;

> p4:=convert(p3,listlist):
> pointplot3d(p4,symbol=circle,symbolsize=30);

with(stats):
y_point:=[seq(-2.95/2+2.95/10*i,i=0..10)];

>k:=1;zzz:=[];
for i from 1 to 11 do
  xx:=[];yy:=[];zz:=[];
  for j from 1 to 11 do
    xx:=[op(xx),p3[k,2]];
    yy:=[op(yy),p3[k,3]];
    k:=k+1;
  end do;
  e1:=fit[leastsquare][[s,t],t=a+b*s+c*s^2+d*s^3+e*s^4,{a,b,c,d,e}]]([xx,yy]);
  f1:=unapply(rhs(e1),s);
  for j from 1 to 11 do
    zz:=[op(zz),f1(y_point[j])];
  end do;
  zzz:=[op(zzz),zz];
end do:

> listcontplot3d(zzz,axes=boxed,orientation=[-60,60],filled=true);

```

# 謝辞

西谷滋人教授には本研究を進めるにあたり細部にわたって有益な助言とご指導  
いただき心から感謝いたします。  
また、同じ研究室の学生の皆様にも感謝いたします。



## 参考文献

- [1] 鈴木秀次,「転位論入門」(アグネ, 東京, 1967)
- [2] 西谷滋人, ”bcc 型金属, 金属間化合物の結合と欠陥構造”(2002)
- [3] MASAKI IGARASHI, ”ATOMISTIC STUDIES ON EXTENDED LATTICE DEFECTS IN HCP METALS”(1991)