

# 54 原子の bcc 鉄に 3d 遷移金属を 固溶した場合の第一原理計算

情報科学科 西谷研究室 4611 河合 真智子

平成 20 年 2 月 21 日

# 概要

bcc 構造 (体心立方格子構造) Fe に 3d 遷移金属が希薄固溶した際の原子半径は、固溶前と比べ、異なる値をとることが実験結果によって明らかにされており、これを有効原子半径と呼ぶ。bcc 鉄の原子半径と、3d 遷移金属の有効原子半径との半径差は、格子配列に歪みを与え、硬度などの物性に大きな影響を与えることとなる。

本研究では、bcc 鉄に各 3d 遷移金属を希薄固溶した際の結晶が持つエネルギーを、VASP を用いた第一原理計算プログラムによって計算する。このように、プログラムを使って計算することで、実験によって実際に値を求めるよりも簡単で、速く結果を得られるという利点がある。ここで求めたエネルギーと体積との関係を E-V (Energy-Volume) 曲線に表し、曲線が与える情報をもとに、各金属結晶の物性予測を試みる。また、有効原子半径と希薄固溶エンタルピーとの関係を調査するとともに、体積弾性率を用いた硬度も曲線から求めることによって、第一原理計算による物性予測が精度の高いものであるかについて検証する。この際、原子数が大きい方が、小さいものより精度が高くなるのではないかについての検証を中心におこなう。研究を進めていく前段階として、結晶の格子配列は物性に大きく関わってくる、すなわち配列の歪みから硬度に変化が生じることを示した。

VASP を使用した計算では、MedeA というソフトを一部使用し、結晶モデルを構築した。結晶モデルは、54 原子で構成された Fe 原子のうち 1 原子を各 3d 元素で固溶した bcc 構造のものとし、さらに、各元素の純粋状態の結晶モデルも構築して、それぞれの E-V 曲線を作成する。この曲線から有効原子半径を求めていき、さらには凝集エネルギー、硬度を求めていく。今回は、54 原子の場合の結果と、前年卒業した前田の研究で求められた 16 原子の場合の結果、さらには実験で求められている結果とを比較して、計算精度について検証した。

# 目次

第1章	緒言	4
第2章	背景	7
2.1	原子半径差と格子配列	7
2.2	硬度と転位	8
2.3	部分モル容積と有効原子半径	9
2.4	凝集エネルギー	10
2.5	体積弾性率 (bulk modulus)	11
第3章	計算方法	12
3.1	VASP による第一原理計算	12
3.2	MedeA を併用した格子モデルの作成	12
3.3	その他の計算ファイルの説明	13
第4章	計算結果	15
4.1	エネルギー-体積曲線	15
4.2	有効原子半径と部分モル容積	19
4.2.1	部分モル容積の導出	19
4.2.2	有効原子半径の導出	20
4.3	凝集エネルギー	26
4.3.1	偏析極限 (segregation limit)	26
4.4	硬度	33
4.5	計算精度の検討	35
第5章	総括	38
第6章	謝辞	39
第7章	附録	41
7.1	k-point1 の Ni の INCAR ファイル	41

7.2	k-point1 の Ni の POSCAR ファイル . . . . .	41
7.3	計算結果 last.txt . . . . .	45
7.4	maple での Fe に 3d 金属を固溶した場合の安定時体積 , エ ネルギーの計算 . . . . .	46
7.5	maple での純粋金属の安定時体積 , エネルギーの計算 . . .	51
7.6	maple での有効原子半径の計算 . . . . .	56
7.7	maple での希薄固溶エンタルピーの計算 . . . . .	59

# 第1章 緒言

20世紀初頭から約40年ほどの間、状態図の研究が金属材料学の主流を占めていた。当時の金属材料の改良と開発は、主として合金法によってなされ、状態図が合金法のただ一つのよりどころであった。このような合金化をすることによって、結晶内に歪みを作り、機械的強度を向上させることができる。産業界においては、合金の強度を向上させることの効果は計り知れないものがあり、熱処理前は比較的加工がし易いことも多大な寄与をする。その中でも、鉄合金は鋼（はがね）という総称として知られ、幅広い産業に大量に用いられるそこで、本研究では、Fe系ナノメタルについて注目したいと思う。鉄という元素は、人間にとって最も利用価値のある元素であると同時に、産業革命以後は産業の中核をなす材料であり「産業の米」などとも呼ばれている。これらのことからわかる通り、Fe系合金において計算機を用いた物性予測が可能となる恩恵は計りしれない。

Fe中に置換固溶した原子は、格子に歪みを与え、マトリックス母相の硬度を変化させる。この量は、転位の動きを通じて、材料全体の強度に影響する。このように、固溶した原子の有効原子半径は材料設計において非常に有用な基礎データである。Fe中に固溶する原子の、固溶前と固溶後の原子半径は図1.1のように計測されている。この図から分かる通り、同じ内殻構造を持つ3d遷移金属系列のTi, V, Cr, Mn, Fe, Co, Ni, Cu, Znは比較的原子半径が近い。そして、他の4d, 5d等の遷移金属系列の際立った系統的な傾向に比べると、それほど系統的な傾向を見いだせない。

近年の第一原理計算法の発展に伴い、格子の歪みの様子や、マトリックスの硬度変化は容易に計算できるようになってきた。このような物性予測の第一段階のプロセスは、第一原理計算による物性のシミュレーションを行う部分であり、密度汎関数法に基づく有効ポテンシャルを用いて

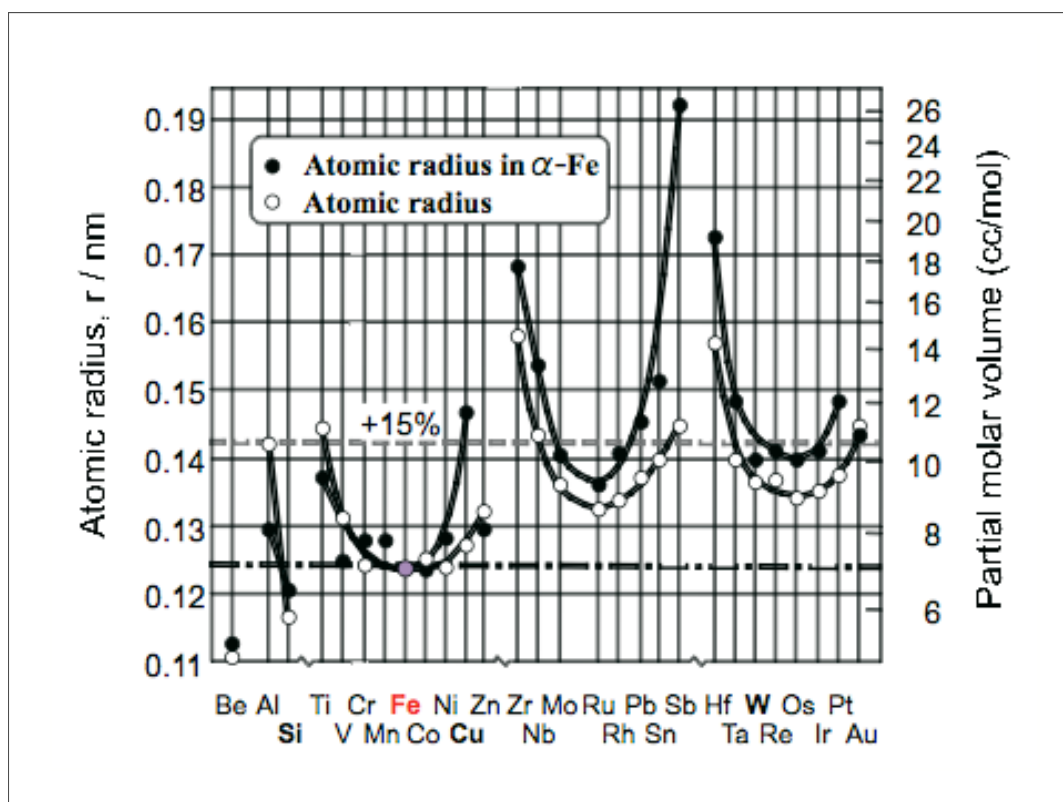
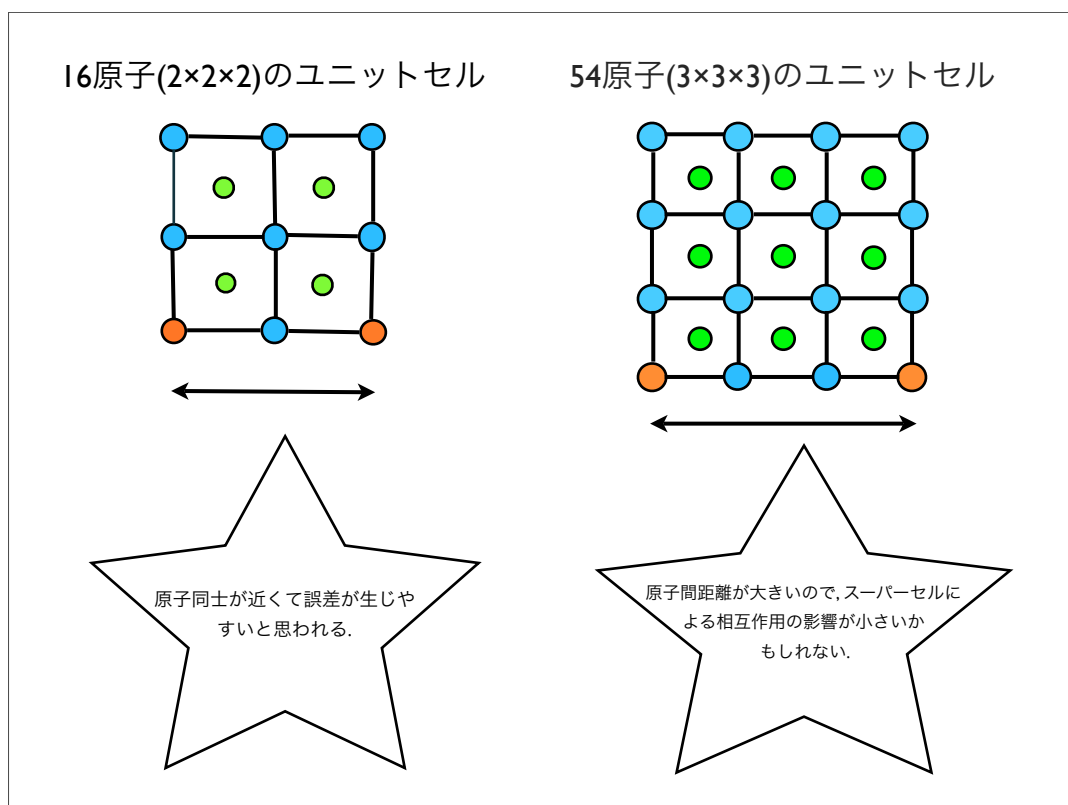


図 1.1: 純粋状態と bcc 鉄への固溶状態の各元素の半径 .

電子状態を決定し、物性予測を行っている。

本研究ではVASP(Vienna Ab-initio Simulation Package) という第一原理計算プログラムを用いて、実際に bcc 鉄に 3d 遷移金属の Ti から Zn までは固溶させる。これらの 2 元合金の全エネルギーを計算し、得られた値を用いて、その E-V 曲線を求めた。この曲線から、原子半径を求め、Fe 中に固溶する原子の固溶前と固溶後の原子半径がどのような挙動を示すかを求めた。さらに Fe-3d 遷移金属系合金における凝集エネルギー、硬度を求め、その系統的な変化から有効原子半径の振る舞いの説明を試みる。また、原子数の多いモデルで計算することで、サイズ効果の影響を抑え、より精度の高い計算結果を得られるのではないかという点から、今までの 16 原子 ( $2 \times 2 \times 2$  のスーパーセル) から、54 原子 ( $3 \times 3 \times 3$  のスーパーセル) に増やして計算することにした。



## 第2章 背景

### 2.1 原子半径差と格子配列

物性予測をしていく上で、結晶モデルの格子配列は非常に重要である。2元合金において、下地となる元素と固溶する元素との原子半径差 ( $\Delta R$ ) が小さい場合は、図 2.1 の左図のような 2 次元の格子配列で表すと、変化はなく、きれいな配列をしている。しかし、主元素と固溶元素との原子半径差 ( $\Delta R$ ) が大きい場合は、図 2.1 の右図のようなふるまいを見せる。格子中でおこるこの「歪み」は、合金の硬度などの性質に大きく関わってくる。

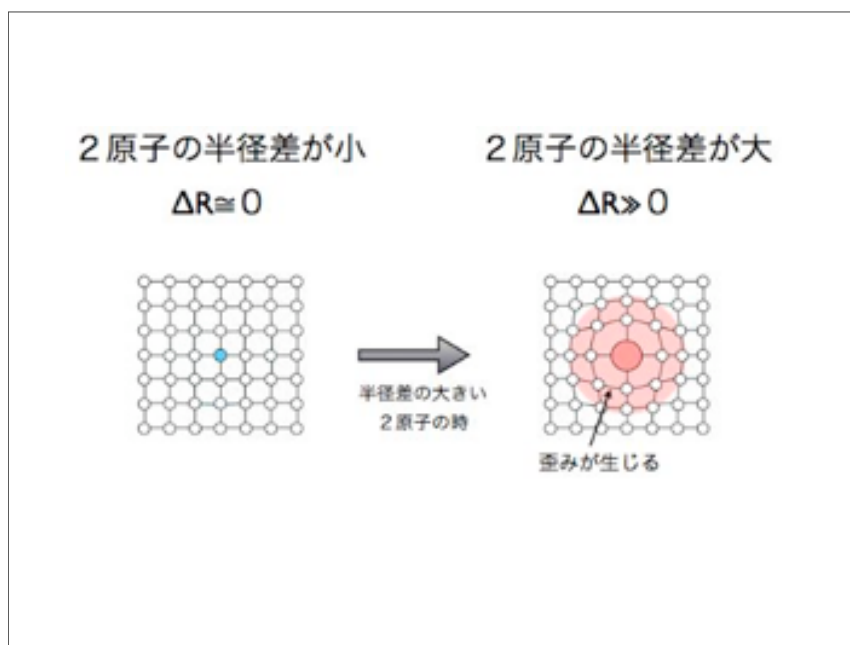


図 2.1: 格子原子と異なった原子が置換した際の、原子半径差による格子の歪みの模式図。



## 2.2 硬度と転位

金属材料学において、硬度はそのものの性質を表すものとして、非常に役立つ要素である。材料の硬度を説明するために、転位と結晶に働く相互作用について説明しておく。結晶に力を加えて変形すると、結晶はそれに応えて同等の力を返す。これを変形応力という。加えた変形量「歪」に対する応力は図 2.2 のような応力歪曲線を示す。

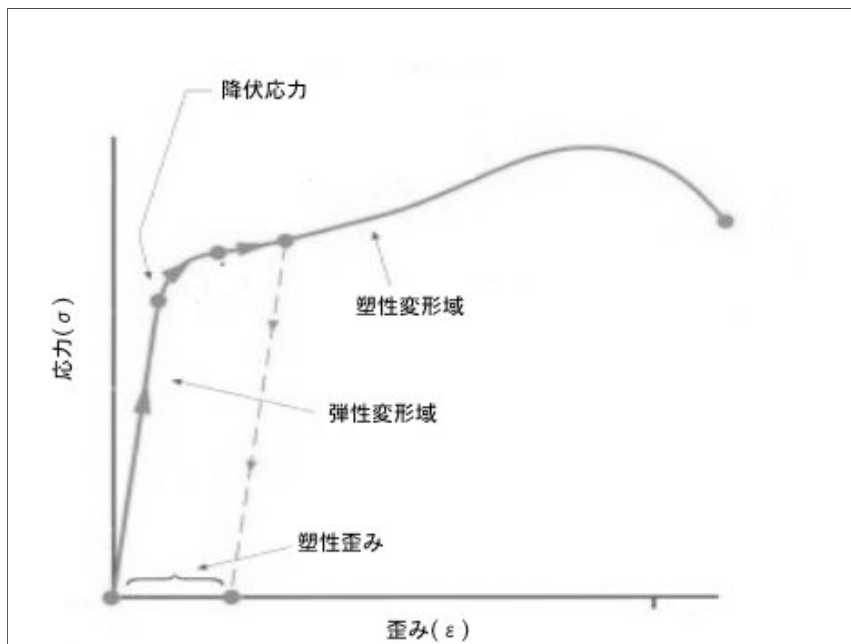


図 2.2: 応力-歪曲線

歪が小さく直線的に応答する領域を弾性変形域、それ以上の変形を加えた領域を塑性変形域と呼ぶ。その限界を降伏応力あるいは降伏強度と呼ぶ。弾性域では変形は格子の相対的は配慮はそのままであるので、加重を除けば元に戻る。しかし、塑性域では、変形がズレによって起こるために、除加重しても永久変形がのこる。つまり、弾性変形域の傾きが大きければ硬い、小さければ軟らかい金属となる。また、領域を長くすれば（降伏応力までの距離が長ければ）、金属の強度は強くなる。このズレをおこす原因が転位である。転位 (dislocation) は材料力学の用語で、結晶中に含まれる、線上の結晶欠陥のことである。転位には刃状転位や、らせん転位、2つが混合した混合転位がある。刃状転位は転位のない結晶に余分面を無理矢理押し込んだ形の結晶欠陥である。らせん転位は転

位線に対して平行に結晶面がずれているものをいう。外力などによって、転位近傍の原子が再配置されることによって転位の位置が移動し材料が変形するため、変形に要する力は原子間の結合力から理論的に計算される力よりも小さく、金属の硬さ（変形のしにくさ）は転位の動きやすさが決めている。転位と溶質原子の相互作用が固溶体においての変形挙動を支配している。転位は溶質原子の近くにとどまっている。転位の動きやすさを支配する機構としては、

転位が固着した原子から離れる際の抵抗

転位が溶質原子を引きずる際の抵抗

のどちらが優勢であるかという結論は決着がついていない。

いずれにせよ、固溶原子が母相とどのように結びついているかが上記の抵抗を支配している。従って、2-1 章、2-2 章で述べたような原子配置での結合の硬度、エネルギーは上記の機構を解明する上で重要な情報を与える。

## 2.3 部分モル容積と有効原子半径

物性に影響を及ぼすものとして、原子の大きさは重要な役割を担っているということがわかった。一般的に、原子の大きさを代表する数値にはいくつかの定義がある。最も単純な値は原子の体積を表すモル容積で、原子 1 個あたりの体積にアボガドロ数をかけて、 $[\text{cc} / \text{mol}]$  で与えられる。この容積を格子モデルに置き換えて、最近接原子間距離を求めることによって、有効原子半径が求まる。本研究における有効原子半径は、bcc 構造 Fe に 3d 元素を固溶した際の原子半径を指す。この有効原子半径と 3d 本来の原子半径を計測した実験データを図 2.3 に示した。縦軸に大きさをとったこの図において、3d 元素群は同じ内殻構造を持ち、Ti ~ Zn の原子半径が近いこともあり、際立った特徴を見いだすことはできない。しかし、詳しく見るとより系統的な振る舞いをしていることがわかる。Fe 付近の V ~ Ni は固溶前も後もほとんど同じである。Ti は固溶前では原子半径が大きい、固溶することで有効原子半径は小さくなる。一方、Cu は固溶前はほぼ同じ原子半径にもかかわらず、固溶後はかなり大きくなる傾向を示している。ところが Zn にこのような傾向は見られない。

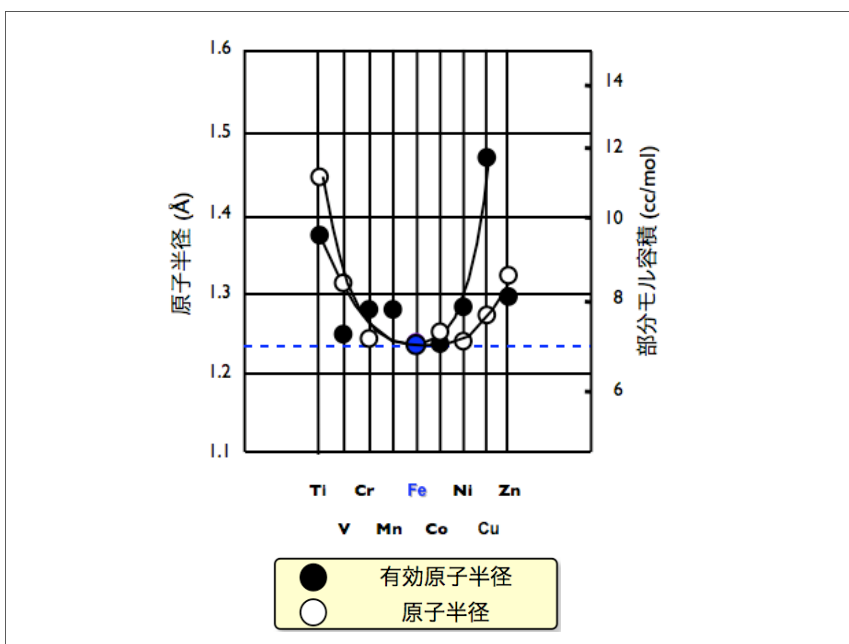


図 2.3: 実験による, bcc Fe に 3d 遷移元素が固溶した場合の有効原子半径 .

縦軸に大きさをとったこの図において, 3d 元素群は同じ内殻構造を持ち, Ti ~ Zn の原子半径が近いこともあり, 際立った特徴を見いだすことはできない. しかし, 詳しく見るとより系統的な振る舞いをしていることがわかる. Fe 付近の V ~ Ni は固溶前も後もほとんど同じである. Ti は固溶前では原子半径が大きい, 固溶することで有効原子半径は小さくなる. 一方, Cu は固溶前はほぼ同じ原子半径にもかかわらず, 固溶後はかなり大きくなる傾向を示している. ところが Zn にこのような傾向は見られない.

## 2.4 凝集エネルギー

2-3 章で示した, 固溶前と固溶後で原子半径が変化するのは理由があり, それは固体をつくる原子は, 固体の環境を好んでいるからである. これが凝集エネルギー (cohesive energy) の起源である. 凝集エネルギーとは, 固体状態にある構成原子を互いに無限に離して相互作用のない孤立原子にするのに必要なエネルギーであり, 正の値をとる. 負の値をとる結合エネルギーの絶対値に等しい. これらの振る舞いは, 固体において

も 2 原子分子の結合，反結合準位の電子占有数と同様に理解することができる．固体では，

$$Eb = \sum_{i=1} E_i n_i \quad (2.1)$$

で示された分子のエネルギー準位と占有数の積の和である結合エネルギーは，エネルギー準位  $E$  と電子状態密度  $n(E)$  の積の積分で，

$$Eb = \sum_{i=1}^n E_i n_i = \int_{-\infty}^{E_F} E n(E) dE \quad (2.2)$$

と求められる．積分範囲はバンドの底あるいは  $-$  からまでである．この項は固体バンド構造によるエネルギーであることから，バンドエネルギーとよばれる．凝集エネルギーが原子の大きさを変化させる要因であると言える．

## 2.5 体積弾性率 (bulk modulus)

また凝集エネルギーは，結晶そのものの硬度にも大きく関わっているということも事実である．本研究において固体の硬度を表現するものとして，体積弾性率を用いた．体積弾性率と硬度には相関があり，系の変形にともなうエネルギー変化が固体の等方的な硬さを表す．体積弾性率が大きい場合，その物質は硬い場合が多い．体積弾性率は以下の式で表すことができる．

$$B_{BM} = \frac{1}{2} \beta \frac{d^2}{d\delta^2} \left( \frac{E}{\Omega} \right) \quad (2.3)$$

この式において  $\beta$  は  $2/9$  として定数で与えられており， $\Omega$  は原子 1 個あたりの体積で表されている．さらに， $B_{BM}$  が GPa 単位で与えられる場合は， $eV/\text{\AA}^3$  への単位換算係数 160.218 で割っておく必要がある．本研究において，体積弾性率は E-V 曲線を用いて表すことが可能であるので，体積弾性率の点からもエネルギーとの関連を検証することは意義深い．

## 第3章 計算方法

### 3.1 VASP による第一原理計算

本研究の主となる要素は、第一原理計算を用いたエネルギー計算である。そのエネルギー計算には VASP を使う。VASP は Wien 工科大学の Hafner 教授が主導して開発した、平面波基底擬ポテンシャル法（ならびに PAW 法）による第一原理計算バンド計算プログラムである。平面波基底の重ね合わせで波動関数を表現し、密度汎関数理論に基づいて電子状態の計算を行う。密度汎関数については、下に少し述べておくこととする。平面波を使用している利点として、その系の原子にかかる力の計算を精確かつ高速に行える点が挙げられ、そのことから VASP は構造最適化や第一原理分子動力学計算のツールとして幅広く用いられている。また、擬ポテンシャル法により、内殻電子をポテンシャルに置き換えて取り扱うので、波動関数の表現に用いる平面波基底の数を大幅に減らし、計算量を軽減する。内殻電子の取り扱いについては、擬ポテンシャル法の他に、全電子計算 PAW 法を採用しており、擬ポテンシャル法と比べさほど計算量を増やすことなく、精度を上げることに成功している。

密度汎関数法とは

密度汎関数理論に基づいた方法で電子系のエネルギーを求める方法と言う。密度汎関数理論とは、与えられた系が縮退のない基底状態を持っている場合、その基底状態が波動関数を用いなくても電子密度を用いて表現できるという理論である。この原理により、基底状態における任意の物理量は電子密度の汎関数で表すことができる。

### 3.2 MedeA を併用した格子モデルの作成

本研究では、VASP をより効率よく使用するために、入力ファイルを簡単に生成する GUI を備えた MedeA というソフトウェアを格子モデルの

構築用として一部利用している．今回，エネルギー計算の基準となる金属結晶はすべて体心立方構造 (bcc) のものであるとした．これは，Fe の金属結晶は bcc 構造が再安定という理由からである．他の原子の安定状態は bcc 構造ではないのだが，Fe 中に含まれた場合の原子の変化を抽出するためにも bcc 構造で統一することが好ましい．また，エネルギー計算において，純粋状態と固溶状態の 2 種類の状態を計算する必要がある．純粋状態は bcc 構造の単位胞である 2 原子からなるモデルを用いたが，計算精度を上げるためにはできるだけ大きなユニットセルを用いることが望まれるので，固溶状態の総原子数は 54 原子とした (53 原子の Fe 原子と，中心原子を他の金属に置換したモデル)．この際，bcc 構造 1 格子辺りに含まれる原子の数は 2 個であるので，3 次元にわたり， $(3 \times 3 \times 3)$  のスーパーセルを用意することで，54 の原子数を持つ格子モデルを再現できる．

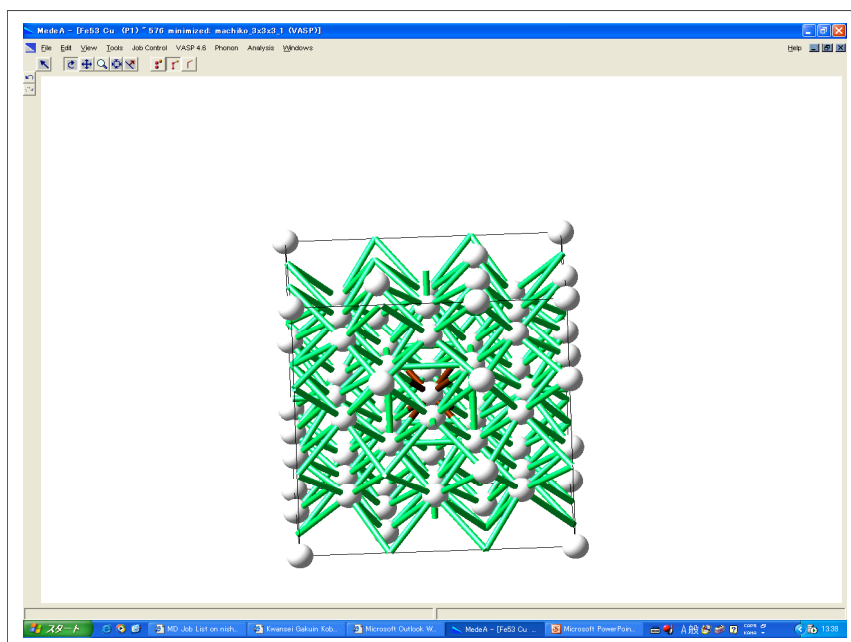


図 3.1: 実際に MedeA で作成した格子モデル.

### 3.3 その他の計算ファイルの説明

VASP の計算を実行するためには，MedeA で作成したモデル以外にも各ファイルを用意しなければならない．それぞれのファイルがどのよう

な入力を担うかを簡単に記しておく．詳しくは，附録 7.1，7.2 章を参照せよ．

#### INCAR

計算を制御するための入力パラメーターが収められているファイル．VASP の中心となるファイルで，VASP をどのように何をさせるか決めるものである．このファイルを扱うときにはエラーや失敗の元となるので気をつけなければならない．

#### POSCAR

計算モデルの格子定数と原子座標が収められている．

#### POTCAR

計算に使用されている各元素のポテンシャルが収められている．大きなファイルなので，閲覧する際は head や grep を使ったほうがよい．

#### KPOINTS

計算に使用される K 点を指定するためのファイル．

#### autocalc

外部緩和における計算を自動でできるシェルスクリプト．

#### res.energy

自動化で計算した際の計算結果が出力されていく．

## 第4章 計算結果

### 4.1 エネルギー-体積曲線

3章で設定した条件下のもと、実際に各3d遷移金属に対して計算を行った。ただし、本研究では、16原子での計算とはファイル INCAR 内での計算精度を変えて行った。ファイルはシェル内で、コマンド `emacs` を使うことで書き換えることができる。

ファイル INCAR(16 原子の場合)

```
# SCF input for VASP
# Note that VASP uses the FIRST occurrence of a keyword
SYSTEM = Cu3.56 (VASP)

PREC = Medium
NBANDS = 300
NELM = 100
NELMIN = 2
EDIFF = 1.0e-05
ISPIN = 2
ICHARG = 2
LWAVE = .FALSE.
LCHARG = .TRUE.
ISMear = 0
SIGMA = 0.1
IALGO = 48
LREAL = .FALSE.
IBRION = 2
ISIF = 2
NFREE = 2
```



```
NSW = 30
```

### ファイル INCAR(54 原子の場合)

```
# SCF input for VASP
# Note that VASP uses the FIRST occurrence of a keyword
SYSTEM = Cu3.56 (VASP)

PREC = High
NBANDS = 400
NELM = 100
NELMIN = 2
EDIFF = 1.0e-05
ISPIN = 2
ICHARG = 2
LWAVE = .FALSE.
LCHARG = .TRUE.
ISMear = 0
SIGMA = 0.1
IALGO = 48
LREAL = .FALSE.
IBRION = 2
ISIF = 2
NFREE = 2
NSW = 50
```

このように、PREC を Medium から High に書き換え、NBANDS を 300 から 400、または 450 に増やした。また、NSW は 30 から 50 に増やしている。これは、54 原子の計算過程では、計算回数が 30 回だと収束しきれない場合があり、計算結果に誤差が生じると分かったからである。

こうして、計算結果によって得られた各ポイント(体積)と全体エネルギーを 2 次元の曲線上に表すために、最小二次近似法を用いて fitting を行い、 $x$  と  $y$  の関数に変換する。今回この関数は 5 次とした(附録 7.4 参照)。ここでは例として、Fe<sub>54</sub> 原子のものと、Fe<sub>53</sub>Ni<sub>1</sub> のものを図示した。

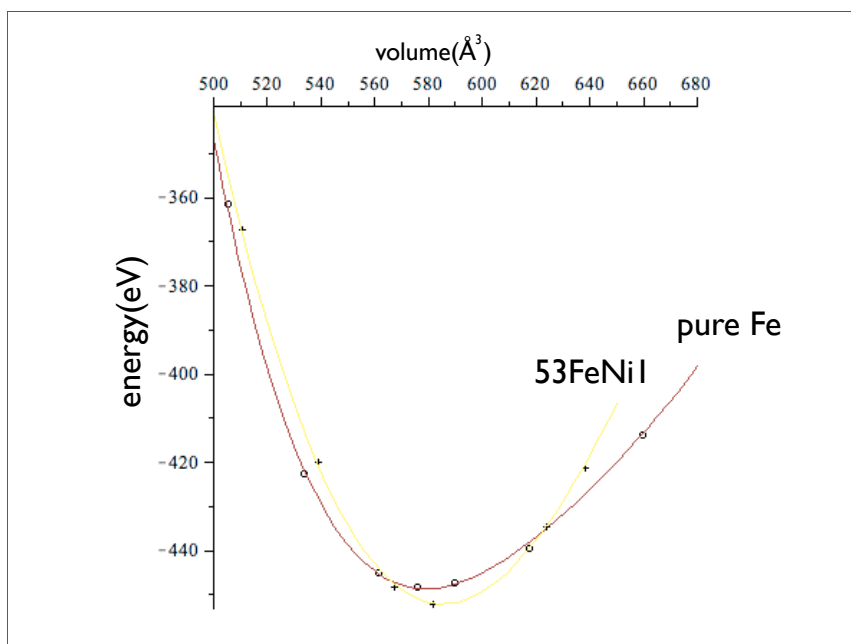


図 4.1: 純粋状態の鉄と Ni 原子が固溶した状態のエネルギー-体積曲線 .

エネルギーの体積依存性を求めた曲線，いわゆる E-V 曲線こそ，第一原理計算の基本ながら重要な情報を与えてくれる．x 軸座標に体積をとり，y 軸座標にエネルギーをとっている．y 軸座標は下にいくほどエネルギーが低くなっている．この図より，各固溶元素と Fe 元素との結合のし易さを測ることができる．無限遠ではお互いが相互作用せず，エネルギー準位としてゼロにとる．そこから距離を近づける（体積を小さくする）と，お互いが引き寄せ合う．さらに，近づけすぎると急激に反発が強くなる．この特徴的な振る舞いは，2 原子間の結合を支配する二つの重要な構成要素を示している．引力と折力（反発の力）がちょうどつり合った位置で，2 原子は平衡を保つ．この曲線においては，頂点の位置にあたる位置が平衡状態であるので，x 座標が各結晶における最安定時の体積となり，y 座標が結合エネルギー（純粋状態の場合は凝集エネルギー）となる．頂点が右方向にいくほど，安定時の体積は大きくなる．この曲線を 1 次微分することによって，最安定時の体積の値はもっと見やすくなる．つまり，y 軸との切片が最安定時の体積となる．

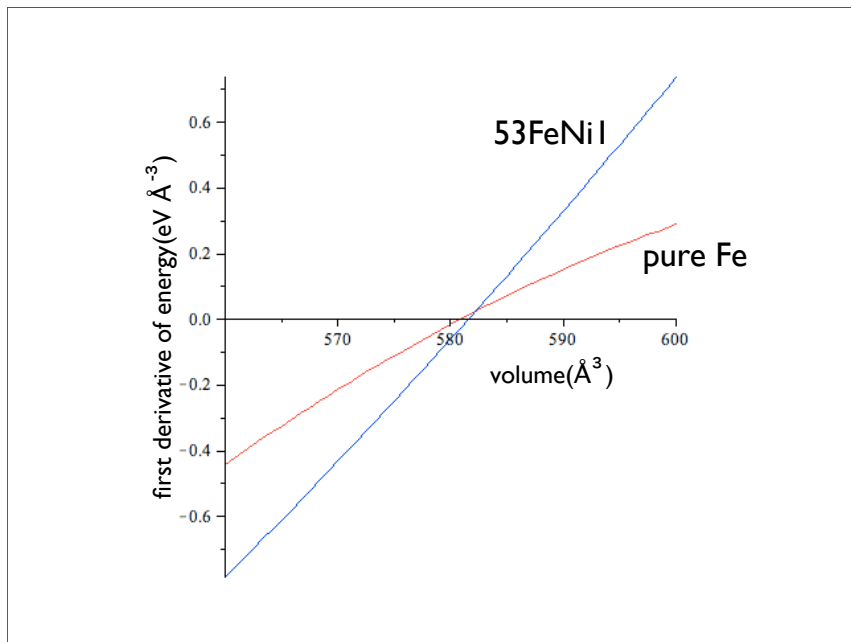


図 4.2: 図 4.1 の曲線を 1 次微分したグラフ .

純 Fe モデルと Ni 固溶モデルを比べると, Ni が固溶することによって体積は膨張しており, 体積弾性率は減少している. このようなエネルギー体積曲線を各 3d 元素について実現することにより, 各 3d 元素が固溶した際の結合エネルギー, 最安定時体積を求めることができる. 図 4.3 に示したものが, 各元素の純粋状態と固溶状態における最安定時体積である.

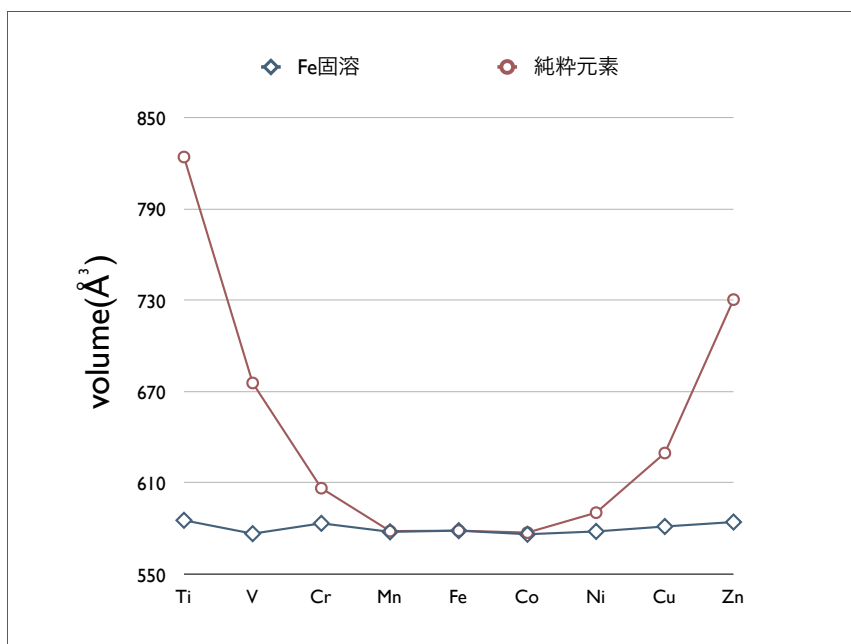


図 4.3: 純粋状態と固溶状態における最安定時体積のグラフ .

このように，E-V 曲線から安定時体積を求めることができ，この値が各元素の部分モル容積や有効原子半径の情報を与えてくれる．4-2 章では，実際に安定時体積から，部分モル容積と有効原子半径を求める．

## 4.2 有効原子半径と部分モル容積

第一原理計算一つを使用するだけで，結晶構造において重要な要素となる原子の大きさを求めるところまでやってくることができた．以下に有効原子半径の値の算出方法を示す．有効原子半径は部分モル容積から求められる．部分モル容積の導出から順を追って説明する．

### 4.2.1 部分モル容積の導出

54 原子で構成された鉄の体積を  $V(Fe)$  とする．計算で出力された再安定時として得た体積  $V(XinFe)$  は，Fe53 原子と固溶元素から構成されたものであるので，全体の体積から Fe53 原子分の体積を取り除けば，固溶元素単体の体積 が求まる．

$$X = V(XinFe) = \frac{53}{54}V(Fe) \quad (4.1)$$

得られた体積の単位は（オングストローム）で表されているので，（立方センチメートル）への単位変換をおこなうと，

$$1 \text{ }^{\circ} = (10^{-8})^3[cc] \quad (4.2)$$

と表すことができ，先ほど得られた  $X$  を，

$$X \text{ }^{\circ} = X^{-24}[cc] \quad (4.3)$$

と変換することができる．この値に 1mol あたりのアボガドロ数 ( $6.02 \times 10^{23}/mol$ ) をかけることで部分モル容積を求めることができる．

$$Vc = X^{-24}[cc] \times 6.02 \times 10^{23}[mol] \quad (4.4)$$

$$Vc = X \times 6.02 \times 10^{-1}[cc/mol] \quad (4.5)$$

$$Vc = 0.602 \times X[cc] \quad (4.6)$$

#### 4.2.2 有効原子半径の導出

上記の各部分モル容積から有効原子半径を求めていく．この際，歪みの影響を全て固溶原子が担っていると仮定する．得られた部分モル容積を 2 倍することによって，bcc 構造における基本単位格子の立方体  $VX$  を再現することができ，この立方体の 3 乗根をとることによって，1 辺あたりの格子定数  $a$  が求まる．

$$VX = Vc \times 2 \quad (4.7)$$

$$VX = a^3 \quad (4.8)$$

次に格子を (1,1,0) 面で切る．こうして隣接原子間距離に変換することによって有効原子半径  $R$  を求める．立方体の対角線は ( $a \times \sqrt{3}$ ) で表す

ことができる．この対角線上に 2 原子が含まれているので，4 原子で割ることによって，原子半径を算出することができる．

$$R = a \times \frac{\sqrt{3}}{4} \quad (4.9)$$

$$R = \sqrt[3]{(VX)} \times \frac{\sqrt{3}}{4} \quad (4.10)$$

$$R = \sqrt[3]{(Vc \times 2)} \times \frac{\sqrt{3}}{4} \quad (4.11)$$

pure 原子においても同様の算出方法で原子半径を求めることによって，当初の目的である 3d 遷移元素の原子半径と bcc Fe に固溶した際の有効原子半径の図を，第一原理計算を用いて再現することができた．

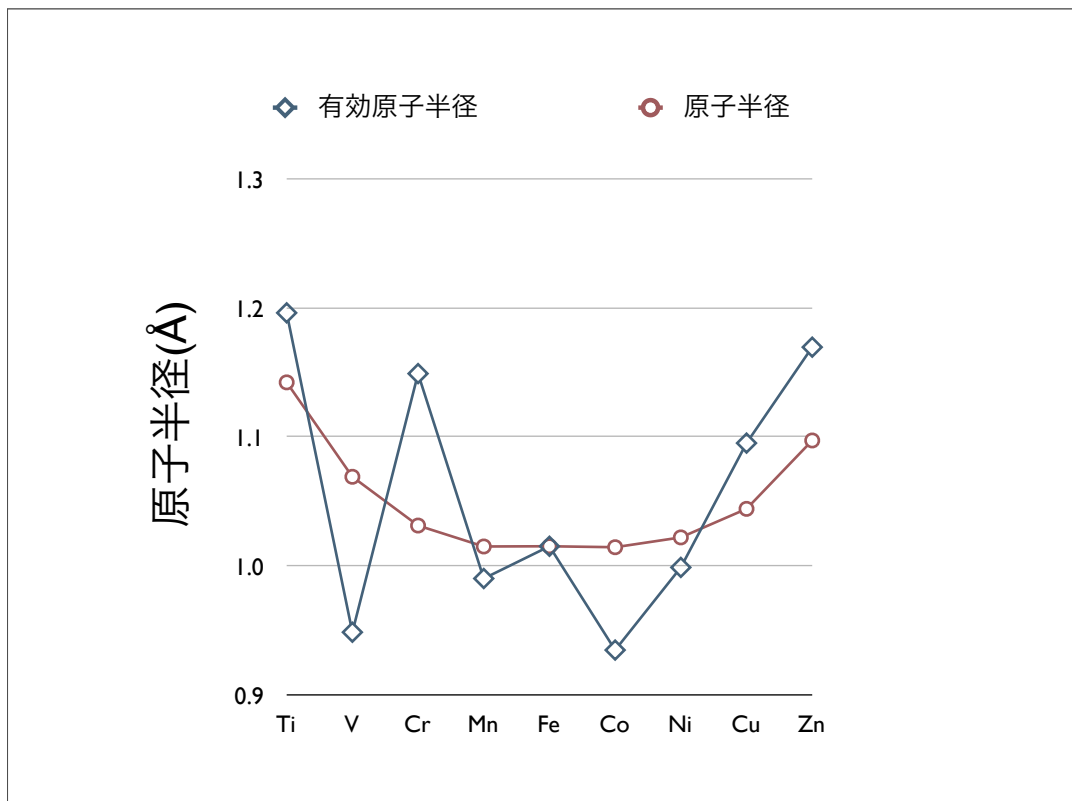


図 4.4: 計算結果による純粋状態の原子半径及び有効原子半径．

しかしながら，図 4.4 で求めた原子半径及び有効原子半径を，16 原子の場合で求められたグラフと比較すると，純粋な原子半径においては，グ

ラフはほぼ同じような形になっているのに比べ，有効原子半径においてはグラフの形が大幅に異なっていることが分かる．また，図 4.4 と実験で得られたグラフを比較しても形が異なっている（下図）．

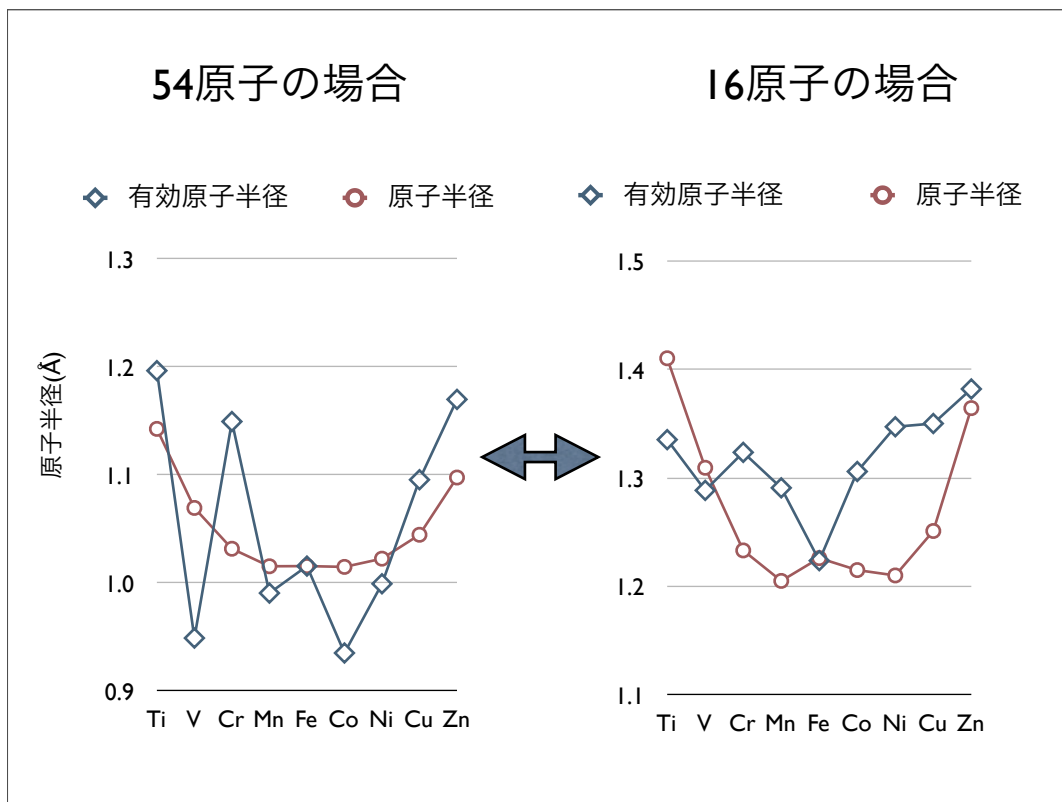


図 4.5: 54 原子の場合と 16 原子の場合の原子半径及び有効原子半径の比較．

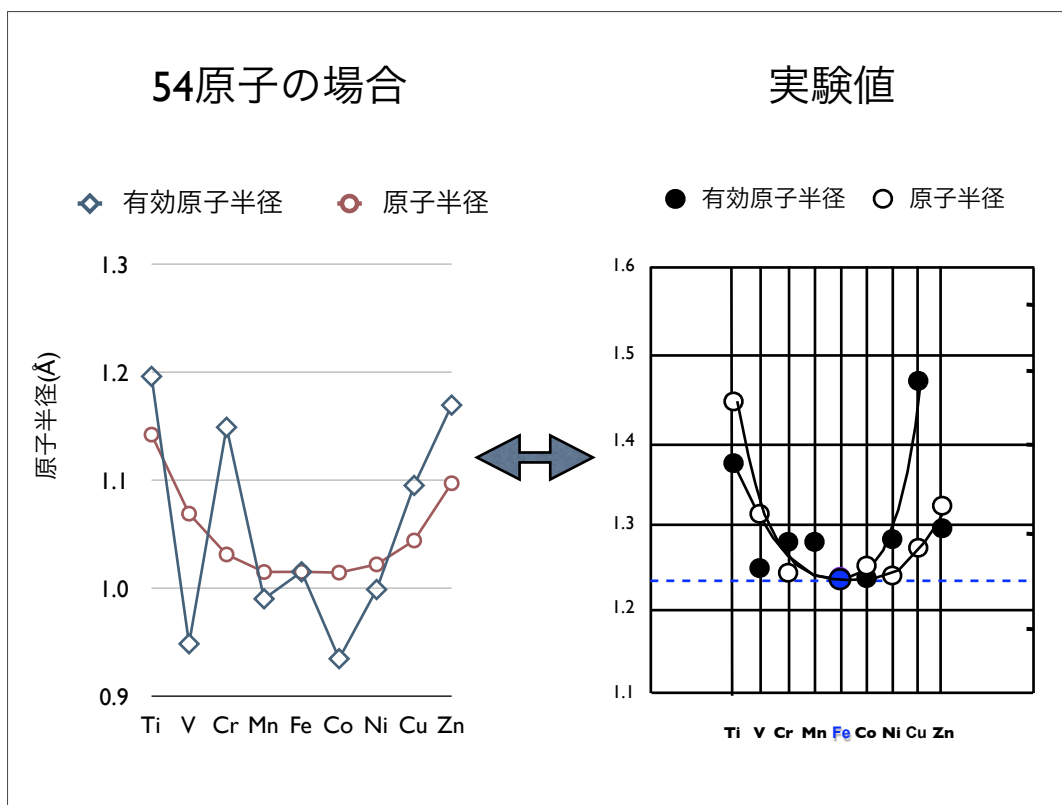


図 4.6: 54 原子の場合と実験での原子半径及び有効原子半径の比較 .

これは、原子数が増えることで、計算の誤差が生じていることが考えられるため、誤差を解消させるために VASP の計算精度を上げることを試みた .

ファイル KPOINTS を、コマンド emacs で書き換える .

ファイル KPOINTS:

Automatic mesh

0

Monkhorst Pack

2 2 2

0. 0. 0.

このファイルを、

Automatic mesh



```

0
Monkhorst Pack
  1  1  1
0.  0.  0.

```

と，書き換える．

このように，数値を 2 から 1 に書き換える．また，1, 3, 5, 7 ... と奇数で書き換えて計算結果を見てみると，k-point=1 で行った場合の計算速度が最も速く，かつ maple で作成したグラフも適切だと思われる．

以上の理由から，特に点の位置が異なっている Ti, V, Co, Ni, Cu の計算を，やり直した．

こうして作成したグラフを図 4.7, 図 4.8 に示した．

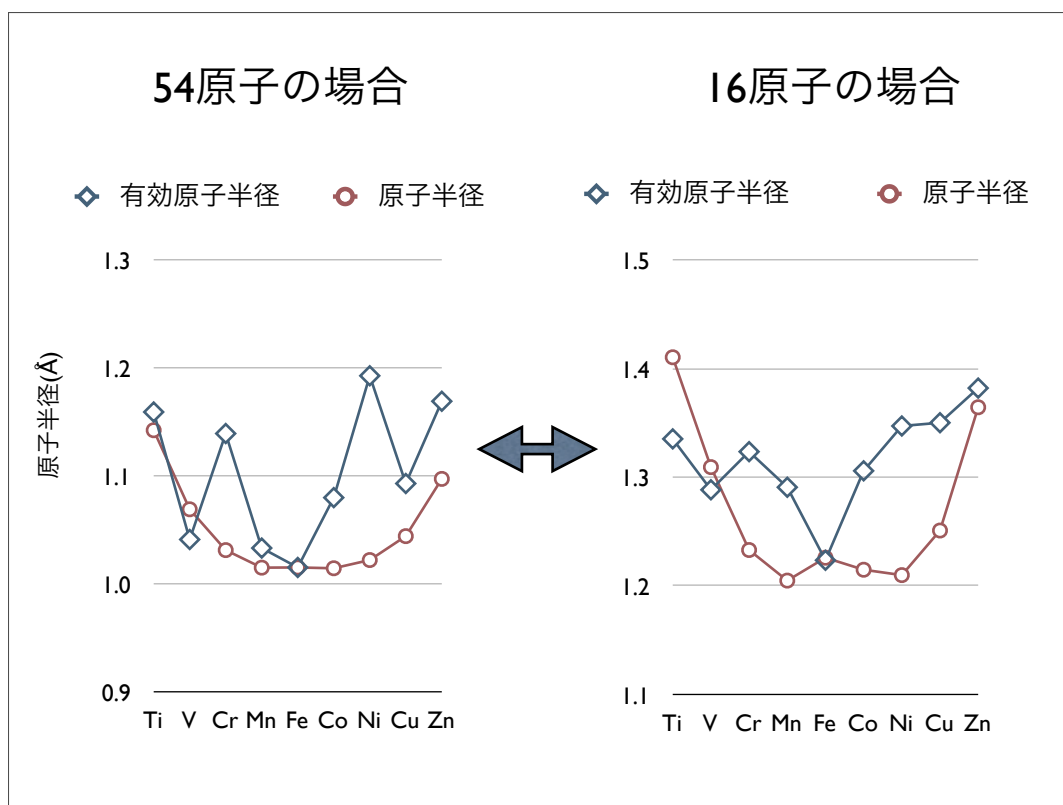


図 4.7: 54 原子の場合と 16 原子の場合の原子半径及び有効原子半径の比較．

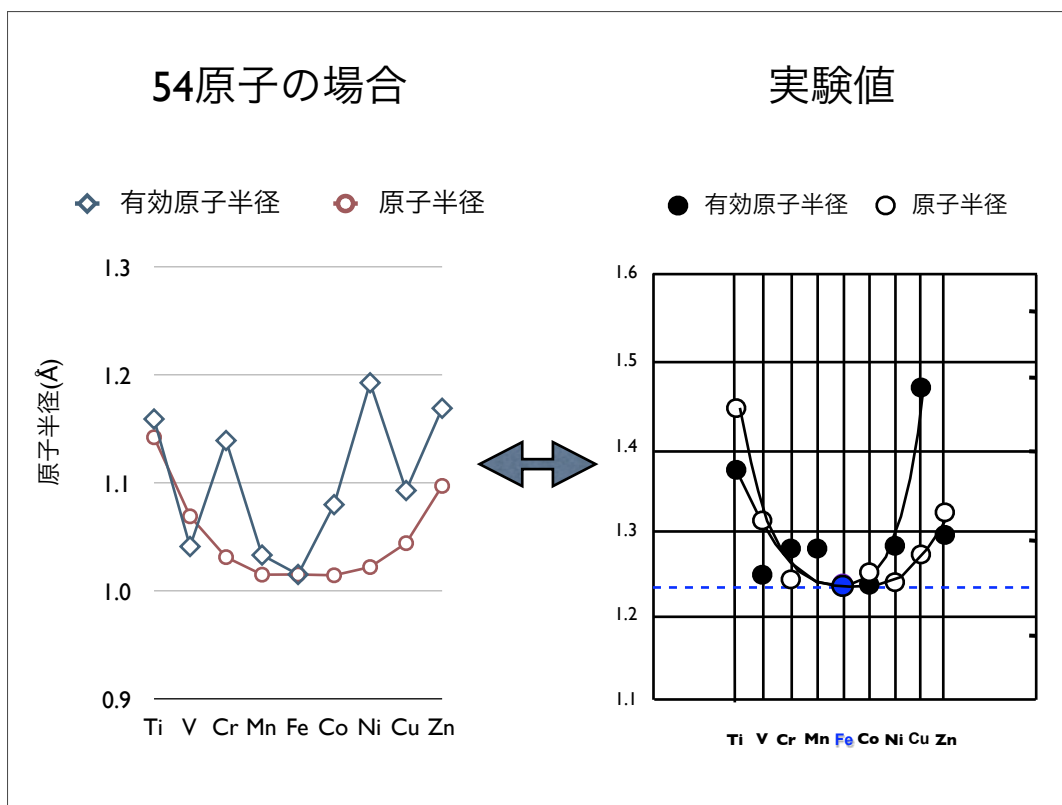


図 4.8: 54 原子の場合と実験による原子半径及び有効原子半径の比較 .

また , k-point=2 の場合と k-point=1 の場合で比較すると , 図 4.9 のようになる .

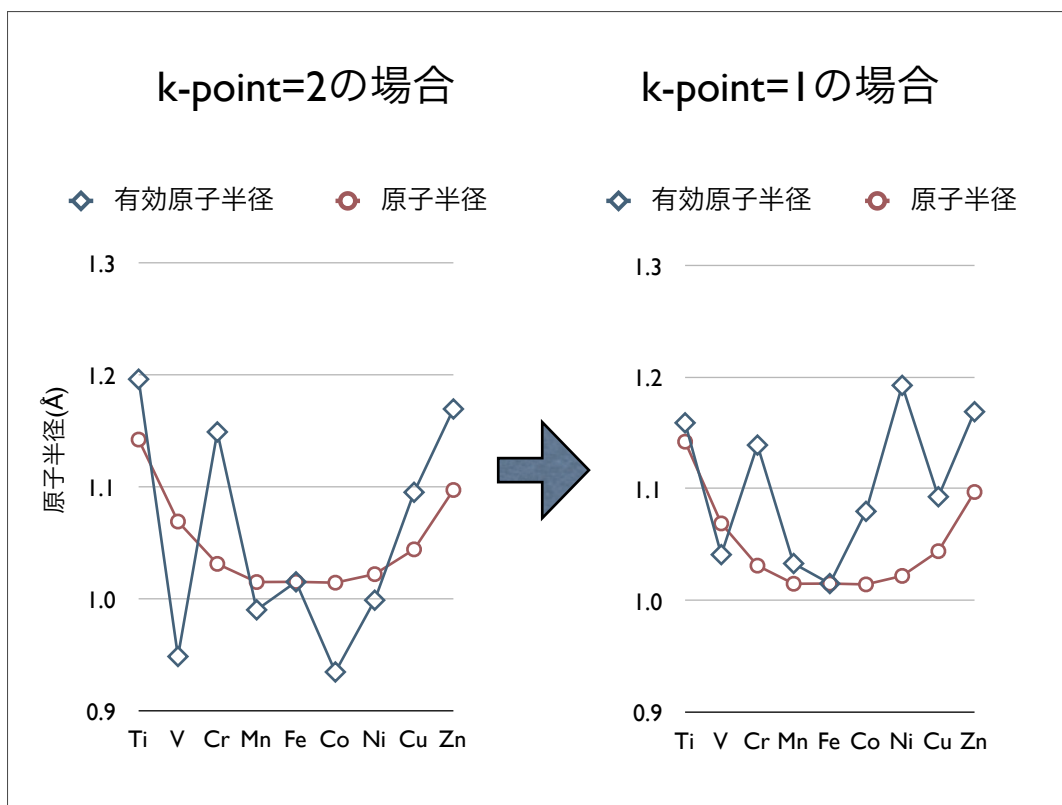


図 4.9: k-point=2 の場合と k-point=1 の場合の原子半径及び有効原子半径の比較．

以上の3つのグラフから，54原子での計算を行う場合は，k-point=1で行う方が16原子での結果や実験での結果と近くなることが分かる．

## 4.3 凝集エネルギー

純金属モデルを固溶金属モデルと直接比べるのは難しい．そこでよく用いられる偏析極限 (segregation limit) を用いる．これは，純金属モデルと純 Fe とのモデル原子数の比を取った平均値あるいは内挿値である．

### 4.3.1 偏析極限 (segregation limit)

自由エネルギーは，モルあたりの示強変数にかえて議論することが通常行われる．多元系でのモルあたりの自由エネルギーを  $G$  とする．単相

A, B の自由エネルギーを  $G_A, G_B$  とする．A, B 原子が混合して 1 mol の溶体をつくるとする．A, B 原子の組成を  $X_A, X_B$  とすると，自由エネルギーがこの 2 点を結んだ直線のととき，すなわち

$$G_S = X_A \times G_A + X_B \times G_B \quad (4.12)$$

は，2 相が完全に分離した状態とみなせる．このような状態は偏析極限 (segregation limit) と呼ばれる．自由エネルギー変化はここからのズレ ( $G - G_S$ ) で測られる．

本研究において，Fe54 原子に各元素を固溶した時の結合エネルギーが  $G$  にあてはまる．そして鉄と各 3d 遷移元素の組成を 53:1 で構成したエネルギーが  $G_S$  にあてはまる．例として，16 原子での偏析極限と希薄極限のイメージ図を図 4.10 に示す．

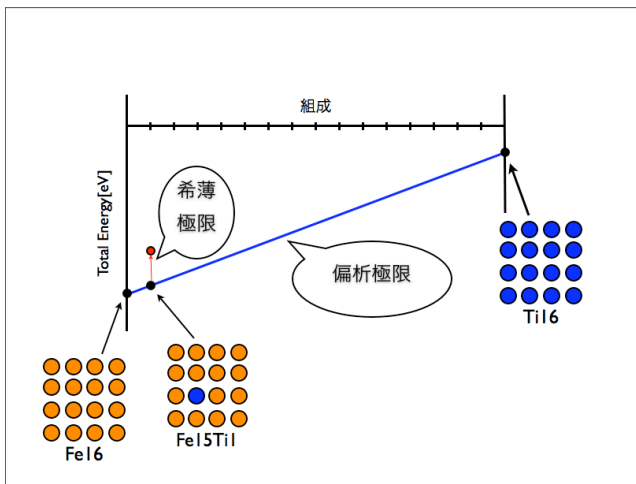


図 4.10: 偏析極限と希薄極限のイメージ図．

また，k-point=2 の場合と k-point=1 の場合の偏析極限と固溶金属モデルの全エネルギーを図 4.11 に，同様に 16 原子での図 4.12 も示す．

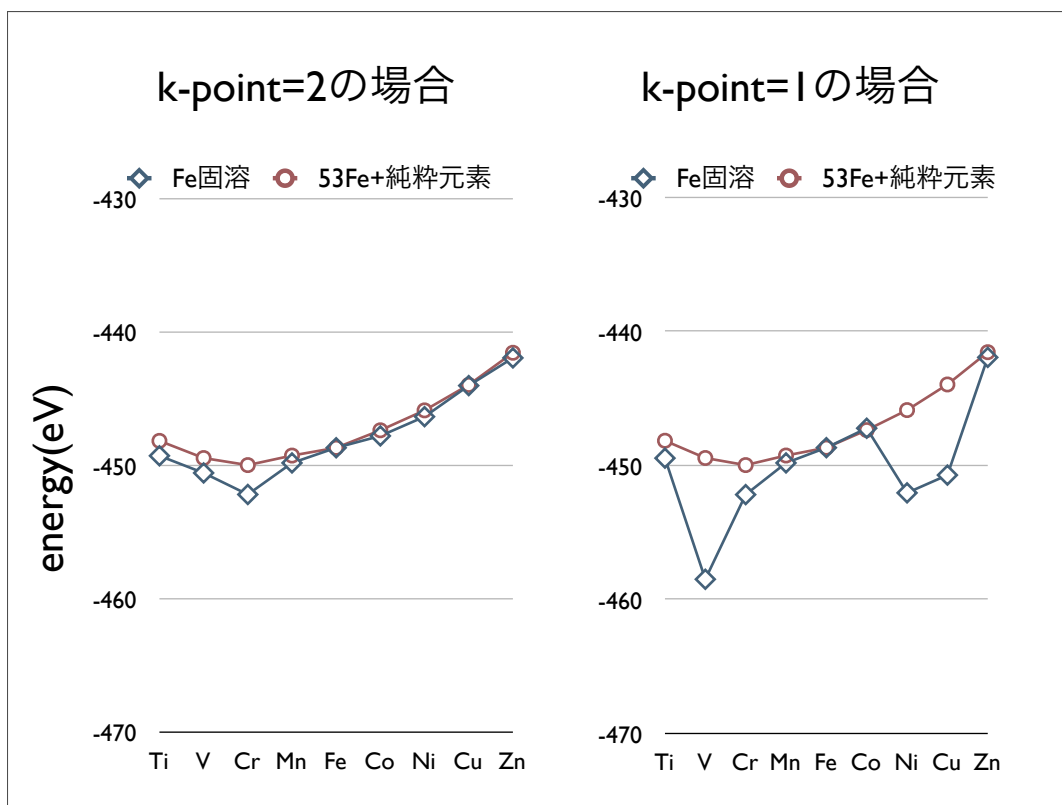


図 4.11: k-point=2 と k-point=1 の偏析極限と固溶金属モデルの全エネルギー図 .

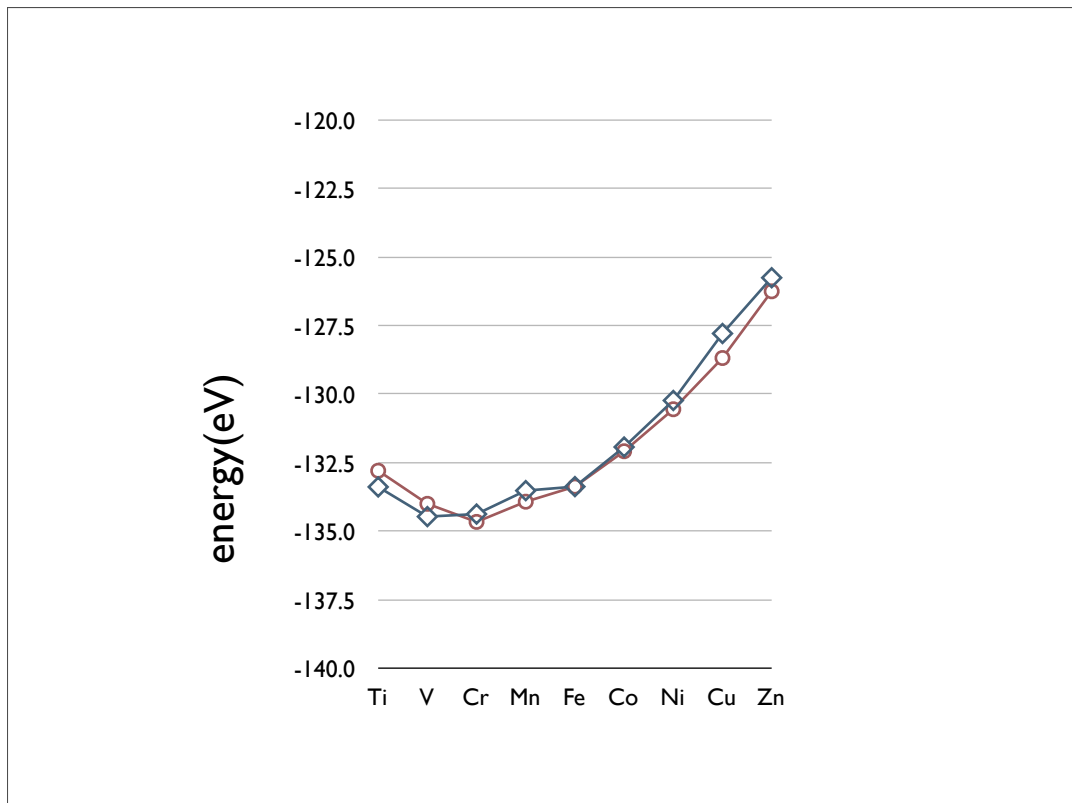


図 4.12: 16 原子での偏析極限と固溶金属モデルの全エネルギー図．

以上の図から， $k\text{-point}=2$  での固溶モデルエネルギーは，16 原子でのエネルギー図と近いのに比べ， $k\text{-point}=1$  での固溶モデルエネルギーは，V，Ni，Cu で大きく異なっている．

また，傾向をさらに読み取りやすくするために，この 2 つのエネルギーの差をとる．この差は希薄極限 (dilution limit) に相当する．これにより，Fe 中に金属元素が希薄極限で固溶した場合の生成エネルギー，つまり希薄固溶エンタルピーに相当する値 ( $\Delta H$ ) を得ることができる．

$$\Delta H = G - G_S \quad (4.13)$$

そして各希薄固溶エンタルピーに相当する値を図 4.3.1 に，同様に 16 原子での値を図 4.3.1 に示す．ただし，実測される希薄極限では，偏析極限は bcc 以外の安定構造がとられるので，その値はここで得られた値からずれている．

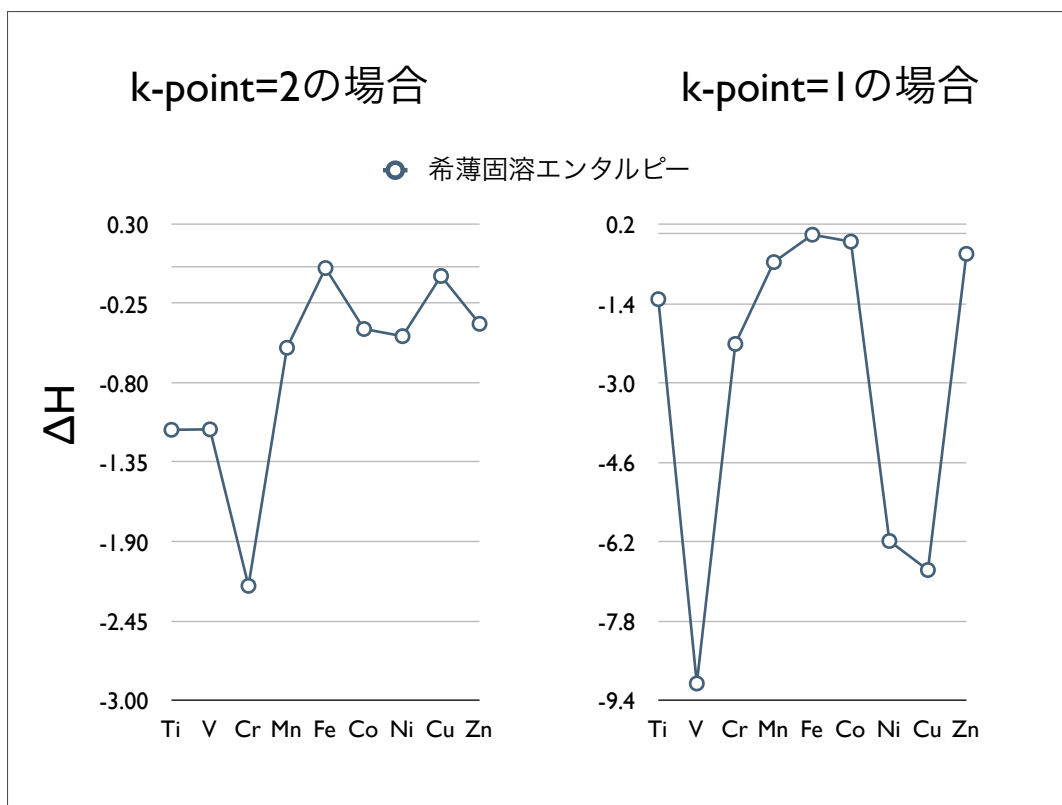


図 4.13: k-point=2 と k-point=1 の希薄固溶エンタルピー図

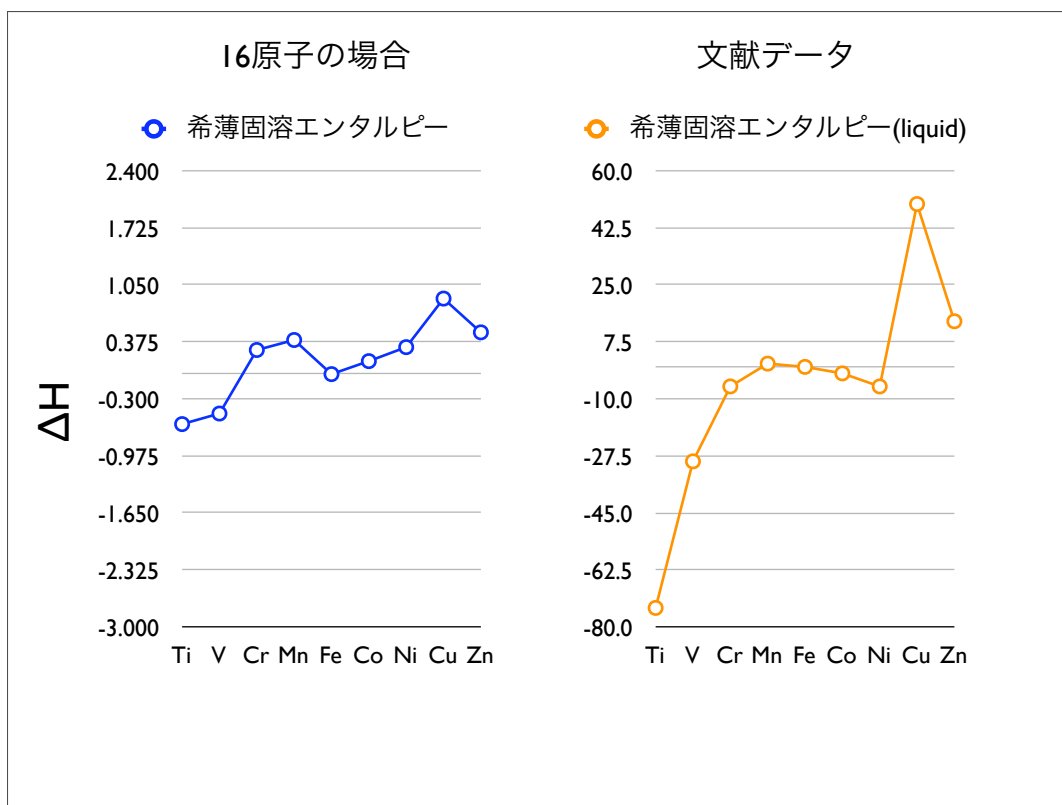


図 4.14: 16 原子と文献データの希薄固溶エンタルピー図

以上の図をまとめると，次のようになる．



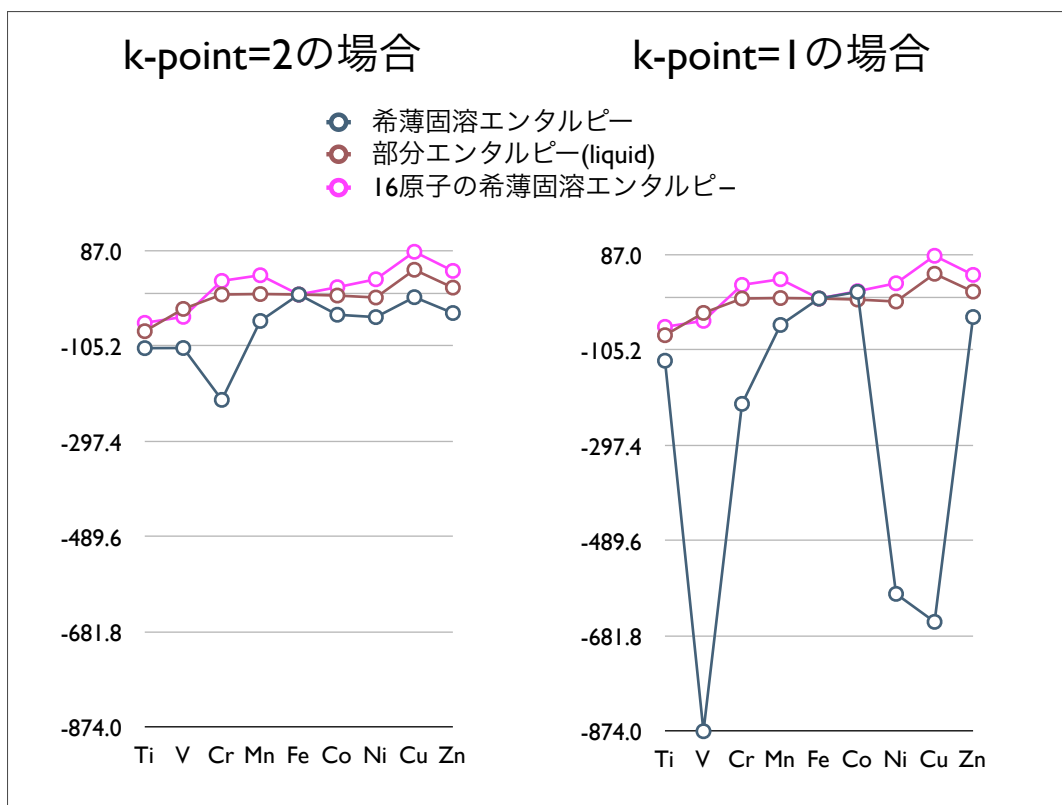


図 4.15: 54 原子，16 原子，文献データの希薄固溶エンタルピー図．

図 4.3.1 から，希薄固溶エンタルピー図においては， $k\text{-point}=2$  の場合の方が， $k\text{-point}=1$  に比べて 16 原子でのエンタルピー図や，文献データのエンタルピー図と近いことが分かる．

このとき，図 4.3.1 では，エンタルピー値が正になっている値がほとんどみられない．これは，54 原子のような大きなユニットセルでの計算では遍析極限や希薄極限を求める際，わずかな差によって値が大きく左右されてしまい，誤差が生じるからだと思われる．

また，このエンタルピー図と，図 4.9 のグラフを比較すると，この傾向と図 4.9 で得られた純金属と固溶金属の原子半径差には負の相関があることが読み取れる．つまり，純金属よりも固溶金属が小さくなる V は希薄固溶エンタルピーは小さく（本来だと負に）なり，他の純金属よりも固溶金属が大きくなる元素では大きく（本来だと正に）なる．これは，古く

から提唱されている『希薄固溶エンタルピーが正，すなわち原子同士が退けあう傾向のある原子同士は半径が大きくなる．逆も同様である』という原理と一致する．この原理は，近年第一原理計算によって簡単に示すことが可能となっている．

## 4.4 硬度

硬度に関しても同様の傾向が見られる．E-V 曲線における V にあたる部分を体積ではなく格子定数 ( $\hat{V}^{1/3}/2$ ) に変更して，格子定数-エネルギー曲線において示した図において体積弾性率で表した硬度を求めることができる．エネルギーがもっとも低い距離を平衡原子間距離，エネルギー値が凝集エネルギーである．この結合エネルギー曲線の 2 次微分 (曲率) から硬度を表す体積弾性率 (bulk modulus) が求まる．(実際に maple で求めた体積弾性率の式  $\text{bulk} := \text{格子定数}^2/9 / (\text{最安定時エネルギー}) * 160.218^2$ ); ．図 4.16 に純金属モデルと固溶モデルの体積弾性率を示した．

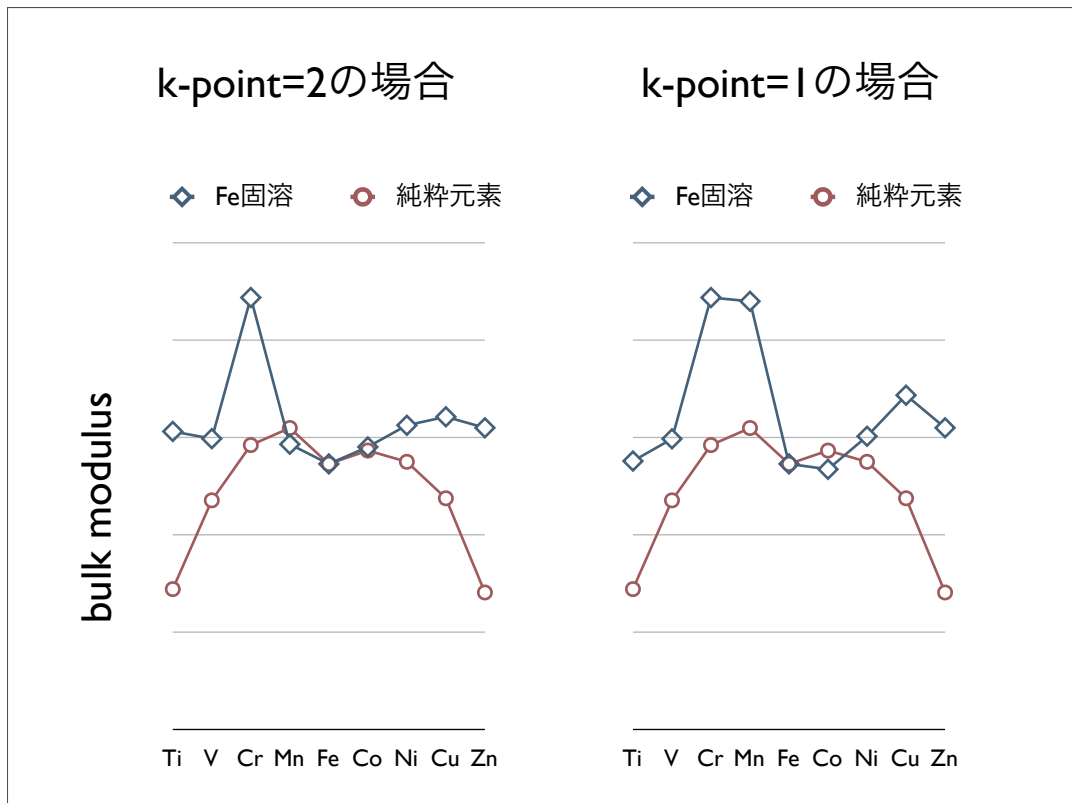


図 4.16: 純粋状態と固溶状態における各元素の体積弾性率 .

この図から分かる通り， $k\text{-point}=2$ での固溶状態の体積弾性率は，希薄固溶エンタルピーの変化と逆向きの動きをしている．つまり，希薄固溶エンタルピーが正の元素は固溶モデルの硬度が減少し，逆に希薄固溶エンタルピーが負の元素は硬度が上昇しているといえる．

希薄固溶エンタルピー	負	正
原子間力	強い	弱い
硬度	上昇する	減少する

## 4.5 計算精度の検討

ここで、最安定時エネルギーのズレがエンタルピーの値にどれくらい影響するのかを計算した。

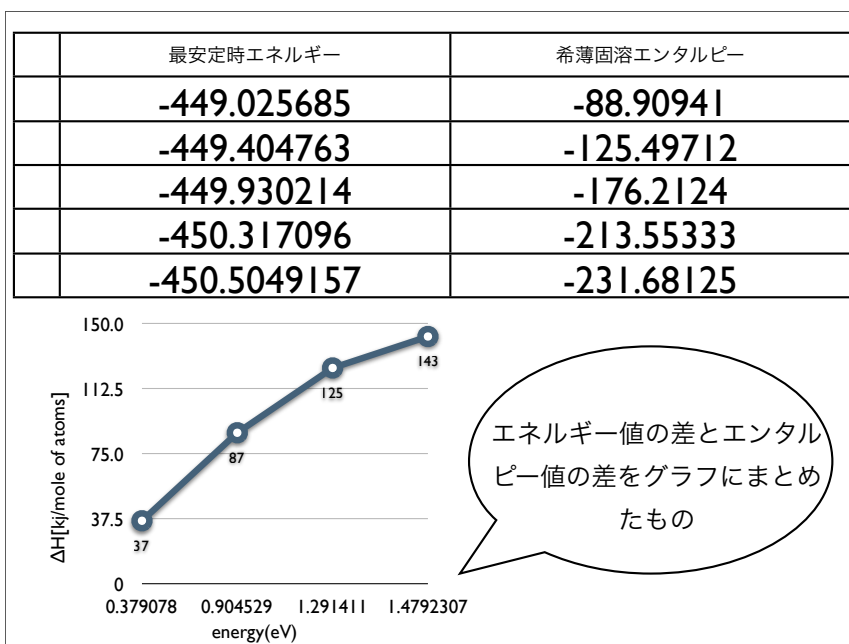


図 4.17: Ti における最安定時エネルギーと希薄固溶エンタルピーについての図。

この表から分かる通り，エネルギー値が約 0.38 ずれると，エンタルピー値が 37 もずれることが分かった．つまり，わずかな差が生じただけで，エンタルピーに大きな影響を与えることが証明できた．

原子数 54 の原子モデルの計算結果を，原子数 16 の原子モデルの計算結果や実験値と比較して，その第一原理計算による計算をさらに正確に計算させることを目的としたが，54 原子という大きなモデルでの計算では計算過程での誤差が生じやすく，精度の高い計算を行うことが困難であった．この中で，計算ファイル内で k-point を 2 から 1 に書き換えることで計算精度を上げることを試みたが，結果，有効原子半径を求める際には k-point=1 の方が，k-point=2 の場合に比べて 16 原子での結果や実験値に近かったものの，希薄固溶エンタルピーを求める際には k-point=2 の方が 16 原子での結果や実験値に近い，というデータが得られた．また誤差に関しては，大きな原子数モデルで計算すると，たとえわずかな誤差が生じただけでも，結果に大きなずれを生じさせることが証明された．これにより，全体的にみると，前田のおこなった 16 原子での研究結果の方が，54 原子の結果よりも精度も高く，信頼できることが分かった．同時

に，本研究によって，k-point の数値が大きいものほど，精度の高い計算がおこなえると分かったので，今後は原子数の少ない，小さな原子モデルで，かつ k-point の数値を 3, 5, 7 ... と大きくしたもので計算を進めていくことが望まれる．

## 第5章 総括

本研究では、第一原理計算を用いた計算機シミュレーションで、bcc 構造  $^{54}\text{Fe}$  に 3d 遷移金属を固溶したときの E-V 曲線を求めることで、有効原子半径、凝集エネルギー、硬度を求めた。その際、計算精度を上げるために、計算ファイルの k-point の数の指定を 1 と 2 で比較した。さらに、 $^{16}\text{Fe}$  や、実験によって得られている結果とも比較した。本研究を通して得られた知見を以下で述べる。

- k-point2 において、希薄固溶エンタルピーと硬度との関係をみると、希薄固溶エンタルピーが正の元素は固溶モデルの体積弾性率が減少し、逆に希薄固溶エンタルピーが負の元素は硬度が上昇している。
- VASP の計算では、k-point の値が大きいほど計算精度が高くなる。
- 54 原子の第一原理計算では、16 原子に比べ誤差が生じやすく、計算時間も多くなることから、VASP を用いた第一原理計算プログラムで各金属結晶の物性予測を試みるには、16 原子の方が信頼がおけるとわかった。

## 第6章 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に深い感謝の意を表します．

また，本研究を進めるにつれ，竹田諒平氏，西川篤史氏，また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．



## 関連図書

- [1] 西谷滋人, 『固体物理の基礎』 ( 森北出版株式会社 2006 ) .
- [2] 関西学院大学, 卒業論文, 2007 年 3 月, 前田宰志, 『Fe と 3d 遷移金属における相互作用の第一原理計算』 .
- [3] 加藤雅治 / 熊井真次 / 尾中 晋, 『材料強度学』 ( 朝倉書店 1999 ) .
- [4] 西沢泰二 『日本金属学会会報』 1973, 194.
- [5] A. van de Walle and G. Ceder, Rev. Mod. Phys., 74 (2002), 11.
- [6] F.R. de Boer, R. Boom, W.C.M.Mattens, A.R. Miedema, and A.K.Niessen, “ Cohesion in Metals, ” (North Holland,1988).
- [7] G. Kresse and J.Hafner, Phys. Rev. B, 47(1993), RC558. G.Kresse, Thesis, Technische Universitat Wien 1993. G. Kresse and J. Furthmuller, Comput. Mater. Sci., 6(1996), 15. G. Kresse and J. Furthmuller, Phys. Rev. B, 54 (1996), 11169.
- [8] MedeA, Materials Design 社製ソフト .

## 第7章 附録

### 7.1 k-point1のNiのINCARファイル

```
# SCF input for VASP
# Note that VASP uses the FIRST occurrence of a keyword
SYSTEM = Cu3.56 (VASP)
```

```
PREC = High
NBANDS = 450
NELM = 100
NELMIN = 2
EDIFF = 1.0e-05
ISPIN = 2
ICHARG = 2
LWAVE = .FALSE.
LCHARG = .TRUE.
ISMear = 0
SIGMA = 0.1
IALGO = 48
LREAL = .FALSE.
IBRION = 2
ISIF = 2
NFREE = 2
NSW = 50
```

### 7.2 k-point1のNiのPOSCARファイル

```
Fe53 Ni (P1) ~ Machiko-new2 (VASP)
```

1.0000000000000000		
8.2764491261376810	-0.0000000023172006	-0.0000000023172006
0.0000001454765337	8.2764491261376810	-0.0000000023172006
0.0000001454765337	0.0000001454765339	8.2764491261376810

53 1

Direct

0.0000000000000000	0.0000000000000000	0.0000000000000000
0.1861895677157271	0.1861895677157271	0.1861895677157271
0.0000000000001110	0.0000000000001110	0.3267217642404680
0.1791633355483949	0.1791633355483949	0.499999999998383
-0.0000000000001110	-0.0000000000001110	0.6732782357595322
0.1861895677164359	0.1861895677164360	0.8138104322828553
0.0000000000001110	0.3267217642404680	0.0000000000001110
0.1791633355483949	0.499999999998383	0.1791633355483949
0.0000000000001780	0.3173335937259158	0.3173335937259158
0.1458248676990967	0.5000000000011878	0.5000000000011878
0.0000000000000000	0.3173335937257380	0.6826664062742624
0.1791633355485566	0.5000000000000000	0.8208366644514431
-0.0000000000001110	0.6732782357595322	-0.0000000000001110
0.1861895677164359	0.8138104322828553	0.1861895677164359
0.0000000000000000	0.6826664062742624	0.3173335937257380
0.1791633355485566	0.8208366644514431	0.5000000000000000
-0.0000000000001780	0.6826664062740845	0.6826664062740845
0.1861895677171450	0.8138104322835641	0.8138104322835641
0.3267217642404680	0.0000000000001110	0.0000000000001110
0.499999999998383	0.1791633355483949	0.1791633355483949
0.3173335937259159	0.0000000000001780	0.3173335937259158
0.5000000000011878	0.1458248676990967	0.5000000000011878
0.3173335937257380	0.0000000000000000	0.6826664062742624
0.5000000000000000	0.1791633355485566	0.8208366644514431
0.3173335937259159	0.3173335937259158	0.0000000000001780
0.5000000000011878	0.5000000000011878	0.1458248676990967
0.3488178821542308	0.3488178821542308	0.3488178821542308
0.3488178821543241	0.3488178821543241	0.6511821178455824
0.4999999999988118	0.4999999999988118	0.8541751323009034

0.3173335937257380	0.6826664062742624	0.0000000000000000
0.5000000000000000	0.8208366644514431	0.1791633355485566
0.3488178821543241	0.6511821178455824	0.3488178821543241
0.499999999998118	0.8541751323009033	0.499999999998118
0.3488178821544176	0.6511821178456762	0.6511821178456762
0.5000000000001617	0.8208366644516055	0.8208366644516055
0.6732782357595322	-0.0000000000001110	-0.0000000000001110
0.8138104322828553	0.1861895677164359	0.1861895677164359
0.6826664062742624	0.0000000000000000	0.3173335937257380
0.8208366644514431	0.1791633355485566	0.5000000000000000
0.6826664062740844	-0.0000000000001780	0.6826664062740845
0.8138104322835641	0.1861895677171449	0.8138104322835641
0.6826664062742624	0.3173335937257380	0.0000000000000000
0.8208366644514431	0.5000000000000000	0.1791633355485566
0.6511821178455824	0.3488178821543241	0.3488178821543241
0.8541751323009033	0.499999999998118	0.499999999998118
0.6511821178456762	0.3488178821544176	0.6511821178456762
0.8208366644516055	0.5000000000001617	0.8208366644516055
0.6826664062740845	0.6826664062740845	-0.0000000000001780
0.8138104322835641	0.8138104322835641	0.1861895677171449
0.6511821178456762	0.6511821178456762	0.3488178821544176
0.8208366644516055	0.8208366644516055	0.5000000000001617
0.6511821178457694	0.6511821178457694	0.6511821178457694
0.8138104322842731	0.8138104322842731	0.8138104322842731
0.5000000000000000	0.5000000000000000	0.5000000000000000

0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00
0.00000000E+00	0.00000000E+00	0.00000000E+00

[illegible]

```

0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00
0.00000000E+00  0.00000000E+00  0.00000000E+00

```

### 7.3 計算結果 last.txt

0.90

```

1 F= -.32774384E+03 E0= -.32768414E+03 d E =-.327744E+03 mag=
0.0011

```

0.9

```

1 F= -.32774384E+03 E0= -.32768414E+03 d E =-.327744E+03 mag=
0.0011

```

0.95

```

1 F= -.38027215E+03 E0= -.38005045E+03 d E =-.380272E+03 mag=
0.0003

```

1.0

```

1 F= -.40952577E+03 E0= -.40937103E+03 d E =-.409526E+03 mag=
0.0010

```

1.025

```

1 F= -.41670407E+03 E0= -.41646557E+03 d E =-.416704E+03 mag=
0.0091

```

1.05

```

1 F= -.42010344E+03 E0= -.41975824E+03 d E =-.420103E+03 mag=
5.5071

```

1.075

```

1 F= -.42029786E+03 E0= -.41996034E+03 d E =-.420298E+03 mag=
11.6492

```

```

1.1
  1 F= -.41762279E+03 E0= -.41727130E+03 d E =-.417623E+03 mag=
    12.3674
1.10
  1 F= -.41762279E+03 E0= -.41727130E+03 d E =-.417623E+03 mag=
    12.3674
1.125
  1 F= -.41248132E+03 E0= -.41209047E+03 d E =-.412481E+03 mag=
    18.8832
1.175
  1 F= -.39819667E+03 E0= -.39798513E+03 d E =-.398197E+03 mag=
    59.5731

```

## 7.4 mapleでのFeに3d金属を固溶した場合の安定時体積，エネルギーの計算

```

limit(f,x=a);
                                     f

with(Maplets[Examples]):
file1:="/Users/sakamotoakira/feTi1":

file2:="/Users/sakamotoakira/feV1":
file3:="/Users/sakamotoakira/feCr":

file4:="/Users/sakamotoakira/feMn":
file5:="/Users/sakamotoakira/feFe":
file6:="/Users/sakamotoakira/feCo1":
file7:="/Users/sakamotoakira/feNi1":
file8:="/Users/sakamotoakira/feCu1":
file9:="/Users/sakamotoakira/feZn":

T1:=readdata(file1,2):
T2:=readdata(file2,2):

```

```

T3:=readdata(file3,2):
T4:=readdata(file4,2):
T5:=readdata(file5,2):
T6:=readdata(file6,2):
T7:=readdata(file7,2):
T8:=readdata(file8,2):
T9:=readdata(file9,2):

with(plots):
p11:=pointplot(T1):
p21:=pointplot(T2):
p31:=pointplot(T3):
p41:=pointplot(T4):
p51:=pointplot(T5,symbol=CIRCLE):
p61:=pointplot(T6):
p71:=pointplot(T7,symbol=CROSS):
p81:=pointplot(T8):
p91:=pointplot(T9):

with(stats):
with(linalg):
data1:=convert(transpose(T1),listlist):
data2:=convert(transpose(T2),listlist):
data3:=convert(transpose(T3),listlist):
data4:=convert(transpose(T4),listlist):
data5:=convert(transpose(T5),listlist):
data6:=convert(transpose(T6),listlist):
data7:=convert(transpose(T7),listlist):
data8:=convert(transpose(T8),listlist):
data9:=convert(transpose(T9),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);

```



```

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):
f2:=unapply(rhs(fit2),x);
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):
f3:=unapply(rhs(fit3),x);
fit4:=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}]](data4):
f4:=unapply(rhs(fit4),x);
fit5:=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}]](data5):
f5:=unapply(rhs(fit5),x);
fit6:=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}]](data6):
f6:=unapply(rhs(fit6),x);
fit7:=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}]](data7):
f7:=unapply(rhs(fit7),x);
fit8:=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}]](data8):
f8:=unapply(rhs(fit8),x);
fit9:=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}]](data9):
f9:=unapply(rhs(fit9),x);

p12:=plot(f1(x),x=500.00..680.00):
p22:=plot(f2(x),x=500.00..640.00, color=blue):
p32:=plot(f3(x),x=500.00..640.00, color=green):
p42:=plot(f4(x),x=500.00..650.00, color=black):
p52:=plot(f5(x),x=500.00..680.00, color=brown):
p62:=plot(f6(x),x=500.00..650.00, color=pink):
p72:=plot(f7(x),x=500.00..650.00, color=yellow):
p82:=plot(f8(x),x=500.00..680.00, color=orange):
p92:=plot(f9(x),x=500.00..680.00, color=gray):

```

```
display(p11,p12,p21,p22,p31,p32,p41,p42,p51,p52,p61,p62,
p71,p72,p81,p82,p91,p92);
```

```
d1:=unapply(diff(f1(x),x),x):
d2:=unapply(diff(f2(x),x),x):
d3:=unapply(diff(f3(x),x),x):
d4:=unapply(diff(f4(x),x),x):
d5:=unapply(diff(f5(x),x),x):
d6:=unapply(diff(f6(x),x),x):
d7:=unapply(diff(f7(x),x),x):
d8:=unapply(diff(f8(x),x),x):
d9:=unapply(diff(f9(x),x),x):
```

```
kai1:=fsolve(d1(x)=0,x=560..600);
kai2:=fsolve(d2(x)=0,x=560..610);
kai3:=fsolve(d3(x)=0,x=560..640);
kai4:=fsolve(d4(x)=0,x=560..640);
kai5:=fsolve(d5(x)=0,x=560..640);
kai6:=fsolve(d6(x)=0,x=560..620);
kai7:=fsolve(d7(x)=0,x=560..620);
kai8:=fsolve(d8(x)=0,x=560..630);
kai9:=fsolve(d9(x)=0,x=560..640);
```

```
584.1330509
582.2862287
583.7200666
```

578.1272040  
578.8994509  
580.8408630  
585.3821685  
581.5537191  
584.5662502

```
dp1:=plot(d1(x),x=560.00..600.00):  
dp7:=plot(d7(x),x=560.00..600.00,color=blue):  
  
display(dp1,dp7);
```

```
y1:=f1(kai1);  
y2:=f2(kai2);  
y3:=f3(kai3);  
y4:=f4(kai4);  
y5:=f5(kai5);  
y6:=f6(kai6);  
y7:=f7(kai7);  
y8:=f8(kai8);  
y9:=f9(kai9);
```

-449.404763  
-458.443123  
-452.12553  
-449.76054  
-448.62308  
-447.172804  
-451.995224  
-450.6686816  
-441.88426

## 7.5 mapleでの純粋金属の安定時体積, エネルギーの計算

```
limit(f,x=a);

with(Maplets[Examples]):
file1:="/Users/sakamotoakira/pTi":
file2:="/Users/sakamotoakira/pV":
file3:="/Users/sakamotoakira/pCr":

file4:="/Users/sakamotoakira/pMn":
file5:="/Users/sakamotoakira/feFe":
file6:="/Users/sakamotoakira/pCo":
file7:="/Users/sakamotoakira/pNi":
file8:="/Users/sakamotoakira/pCu":
file9:="/Users/sakamotoakira/pZn":

T1:=readdata(file1,2):
T2:=readdata(file2,2):
T3:=readdata(file3,2):
T4:=readdata(file4,2):
T5:=readdata(file5,2):
T6:=readdata(file6,2):
T7:=readdata(file7,2):
T8:=readdata(file8,2):
T9:=readdata(file9,2):
with(plots):
p11:=pointplot(T1):
p21:=pointplot(T2):
p31:=pointplot(T3):
p41:=pointplot(T4):
p51:=pointplot(T5):
p61:=pointplot(T6):
p71:=pointplot(T7):
p81:=pointplot(T8):
```

```

p91:=pointplot(T9):
with(stats):
with(linalg):
data1:=convert(transpose(T1),listlist):
data2:=convert(transpose(T2),listlist):
data3:=convert(transpose(T3),listlist):
data4:=convert(transpose(T4),listlist):
data5:=convert(transpose(T5),listlist):
data6:=convert(transpose(T6),listlist):
data7:=convert(transpose(T7),listlist):
data8:=convert(transpose(T8),listlist):
data9:=convert(transpose(T9),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);
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):
f2:=unapply(rhs(fit2),x);
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):
f3:=unapply(rhs(fit3),x);
fit4:=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}]](data4):
f4:=unapply(rhs(fit4),x);
fit5:=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}]](data5):
f5:=unapply(rhs(fit5),x);
fit6:=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}]](data6):
f6:=unapply(rhs(fit6),x);
fit7:=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}]](data7):
f7:=unapply(rhs(fit7),x);
fit8:=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}]](data8):

```

```

f8:=unapply(rhs(fit8),x);
fit9:=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}]](data9):
f9:=unapply(rhs(fit9),x);
p12:=plot(f1(x),x=600.00..920.00):
p22:=plot(f2(x),x=520.00..800.00, color=blue):
p32:=plot(f3(x),x=500.00..720.00, color=green):
p42:=plot(f4(x),x=500.00..700.00, color=black):
p52:=plot(f5(x),x=520.00..680.00, color=brown):
p62:=plot(f6(x),x=500.00..700.00, color=pink):
p72:=plot(f7(x),x=500.00..700.00, color=yellow):
p82:=plot(f8(x),x=520.00..760.00, color=orange):
p92:=plot(f9(x),x=600.00..820.00, color=gray):

display(p11,p12,p21,p22,p31,p32,p41,p42,p51,p52,
p61,p62,p71,p72,p81,p82,p91,p92);

```

```

d1:=unapply(diff(f1(x),x),x):
d2:=unapply(diff(f2(x),x),x):
d3:=unapply(diff(f3(x),x),x):
d4:=unapply(diff(f4(x),x),x):
d5:=unapply(diff(f5(x),x),x):
d6:=unapply(diff(f6(x),x),x):
d7:=unapply(diff(f7(x),x),x):
d8:=unapply(diff(f8(x),x),x):
d9:=unapply(diff(f9(x),x),x):

kai1:=fsolve(d1(x)=0,x=700..900);
kai2:=fsolve(d2(x)=0,x=600..700);
kai3:=fsolve(d3(x)=0,x=500..700);
kai4:=fsolve(d4(x)=0,x=560..640);
kai5:=fsolve(d5(x)=0,x=560..640);
kai6:=fsolve(d6(x)=0,x=500..650);

```

```
kai7:=fsolve(d7(x)=0,x=560..640);
kai8:=fsolve(d8(x)=0,x=560..640);
kai9:=fsolve(d9(x)=0,x=700..900);
```

```
824.4153240
675.9562473
606.7953898
578.6080095
578.8994509
577.6273337
590.7221998
629.9118193
730.7265264
```

```
>
y1:=f1(kai1);
y2:=f2(kai2);
y3:=f3(kai3);
y4:=f4(kai4);
y5:=f5(kai5);
y6:=f6(kai6);
y7:=f7(kai7);
y8:=f8(kai8);
y9:=f9(kai9);
```

```
-420.620423
-490.195845
-518.8013127
-480.210791
-448.62308
-377.8295525
-297.163884
```

-193.916805

-63.859137

```
d1:=unapply(diff(f1(x),x),x):
d2:=unapply(diff(f2(x),x),x):
d3:=unapply(diff(f3(x),x),x):
d4:=unapply(diff(f4(x),x),x):
d5:=unapply(diff(f5(x),x),x):
d6:=unapply(diff(f6(x),x),x):
d7:=unapply(diff(f7(x),x),x):
d8:=unapply(diff(f8(x),x),x):
d9:=unapply(diff(f9(x),x),x):
```

```
kai1:=fsolve(d1(x)=0,x=700..900);
kai2:=fsolve(d2(x)=0,x=600..700);
kai3:=fsolve(d3(x)=0,x=500..700);
kai4:=fsolve(d4(x)=0,x=560..640);
kai5:=fsolve(d5(x)=0,x=560..640);
kai6:=fsolve(d6(x)=0,x=500..650);
kai7:=fsolve(d7(x)=0,x=560..640);
kai8:=fsolve(d8(x)=0,x=560..640);
kai9:=fsolve(d9(x)=0,x=700..900);
```

824.4153240

675.9562473

606.7953898

578.6080095

578.8994509

577.6273337

590.7221998

629.9118193

730.7265264

```
dp1:=plot(d1(x),x=100.00..800.00):
```



```

dp8:=plot(d8(x),x=100.00..800.00,color=blue):

display(dp1,dp8);

```

## 7.6 maple での有効原子半径の計算

```

koyou1:=584.1330509:
koyou2:=582.2862287:
koyou3:=583.7200666:
koyou4:=578.1272040:
koyou5:=578.8994509:
koyou6:=580.840863:
koyou7:=578.3888790:
koyou8:=581.5537191:
koyou9:=588.4411771:

```

```

jyun1:=824.4153240:
jyun2:=675.9562473:
jyun3:=606.7953898:
jyun4:=578.6080095:
jyun5:=578.8994509:
jyun6:=577.6273337:
jyun7:=590.7221998:
jyun8:=629.9118193:
jyun9:=730.7265264:

```

```

X1:=koyou1-(53/54)*jyun5;
X2:=koyou2-(53/54)*jyun5;
X3:=koyou3-(53/54)*jyun5;
X4:=koyou4-(53/54)*jyun5;

```

```

X5:=koyou5-(53/54)*jyun5;
X6:=koyou6-(53/54)*jyun5;
X7:=koyou7-(53/54)*jyun5;
X8:=koyou8-(53/54)*jyun5;
X9:=koyou9-(53/54)*jyun5;

15.9539602
14.1071380
15.5409759
9.9481133
10.7203602
12.6617723
10.2097883
13.3746284
20.2620864

Vc1:=0.602*X1;
Vc2:=0.602*X2;
Vc3:=0.602*X3;
Vc4:=0.602*X4;
Vc5:=0.602*X5;
Vc6:=0.602*X6;
Vc7:=0.602*X7;
Vc8:=0.602*X8;
Vc9:=0.602*X9;

9.604284040
8.492497076
9.355667492
5.988764207
6.453656840
7.622386925
6.146292557
8.051526297
12.19777601

R1:=evalf(((Vc1*2)^(1/3))*sqrt(3)/4);
R2:=evalf(((Vc2*2)^(1/3))*sqrt(3)/4);
R3:=evalf(((Vc3*2)^(1/3))*sqrt(3)/4);

```

```

R4:=evalf(((Vc4*2)^(1/3))*sqrt(3)/4);
R5:=evalf(((Vc5*2)^(1/3))*sqrt(3)/4);
R6:=evalf(((Vc6*2)^(1/3))*sqrt(3)/4);
R7:=evalf(((Vc7*2)^(1/3))*sqrt(3)/4);
R8:=evalf(((Vc8*2)^(1/3))*sqrt(3)/4);
R9:=evalf(((Vc9*2)^(1/3))*sqrt(3)/4);

1.159664333
1.113069895
1.149570368
0.9907324157
1.015732287
1.073678603
0.9993440806
1.093461190
1.255849276

x1:=jyun1*1/54:
x2:=jyun2*1/54:
x3:=jyun3*1/54:
x4:=jyun4*1/54:
x5:=jyun5*1/54:
x6:=jyun6*1/54:
x7:=jyun7*1/54:
x8:=jyun8*1/54:
x9:=jyun9*1/54:
vc1:=0.602*x1;
vc2:=0.602*x2;
vc3:=0.602*x3;
vc4:=0.602*x4;
vc5:=0.602*x5;
vc6:=0.602*x6;
vc7:=0.602*x7;
vc8:=0.602*x8;
vc9:=0.602*x9;

9.190704165

```

```

7.535660385
6.764644899
6.450407810
6.453656840
6.439475093
6.585458600
7.022350281
8.146247571

r1:=evalf(((vc1*2)^(1/3))*sqrt(3)/4);
r2:=evalf(((vc2*2)^(1/3))*sqrt(3)/4);
r3:=evalf(((vc3*2)^(1/3))*sqrt(3)/4);
r4:=evalf(((vc4*2)^(1/3))*sqrt(3)/4);
r5:=evalf(((vc5*2)^(1/3))*sqrt(3)/4);
r6:=evalf(((vc6*2)^(1/3))*sqrt(3)/4);
r7:=evalf(((vc7*2)^(1/3))*sqrt(3)/4);
r8:=evalf(((vc8*2)^(1/3))*sqrt(3)/4);
r9:=evalf(((vc9*2)^(1/3))*sqrt(3)/4);

1.142773683
1.069591002
1.031792348
1.015561805
1.015732287
1.014987726
1.022600442
1.044731761
1.097732451

```

## 7.7 mapleでの希薄固溶エンタルピーの計算

#k-point1の最安定時エネルギー

```

y1:=-449.404763:
y2:=-458.443123:
y3:=-452.12553:
y4:=-449.76054:

```

```
y5:=-448.62308:
y6:=-447.172804:
y7:=-451.995224:
y8:=-450.6686816:
y9:=-441.88426:
```

#### #k-point2 の最安定時エネルギー

```
y11:=-449.225741:
y12:=-450.511178:
y13:=-452.12553:
y14:=-449.76054:
y15:=-448.62308:
y16:=-447.734513:
y17:=-446.290451:
y18:=-443.961254:
y19:=-441.88426:
```

#### #純粋元素の最安定時エネルギー

```
py1:=-420.620423:
py2:=-490.195845:
py3:=-518.8013127:
py4:=-480.210791:
py5:=-448.62308:
py6:=-377.8295525:
py7:=-297.163884:
py8:=-193.916805:
py9:=-63.859137:
```

#### #偏析エネルギー

```
a1:=py1*(1/54)+py5*(53/54);
```

```

a2:=py2*(1/54)+py5*(53/54);
a3:=py3*(1/54)+py5*(53/54);
a4:=py4*(1/54)+py5*(53/54);
a5:=py5*(1/54)+py5*(53/54);
a6:=py6*(1/54)+py5*(53/54);
a7:=py7*(1/54)+py5*(53/54);
a8:=py8*(1/54)+py5*(53/54);
a9:=py9*(1/54)+py5*(53/54);

```

```

-448.1045123
-449.3929460
-449.9226769
-449.2080376
-448.62308
-447.3120888
-445.8182801
-443.9062971
-441.4978218

```

#希薄固溶エネルギー

```

y1-a1;
y2-a2;
y3-a3;
y4-a4;
y5-a5;
y6-a6;
y7-a7;
y8-a8;
y9-a9;

```

```

-1.3002507
-9.0501770
-2.2028531
-0.5525024
0.

```

	0.1392848
	-6.1769439
	-6.7623845
	-0.3864382
y11-a1;	
y12-a2;	
y13-a3;	
y14-a4;	
y15-a5;	
y16-a6;	
y17-a7;	
y18-a8;	
y19-a9;	
	-1.1212287
	-1.1182320
	-2.2028531
	-0.5525024
	0.
	-0.4224242
	-0.4721709
	-0.0549569
	-0.3864382

# H を kj に換算

(y1-a1)*2.306*10 <sup>4</sup> *4.1855/1000;	
(y2-a2)*2.306*10 <sup>4</sup> *4.1855/1000;	
(y3-a3)*2.306*10 <sup>4</sup> *4.1855/1000;	
(y4-a4)*2.306*10 <sup>4</sup> *4.1855/1000;	
(y5-a5)*2.306*10 <sup>4</sup> *4.1855/1000;	
(y6-a6)*2.306*10 <sup>4</sup> *4.1855/1000;	
(y7-a7)*2.306*10 <sup>4</sup> *4.1855/1000;	
(y8-a8)*2.306*10 <sup>4</sup> *4.1855/1000;	
(y9-a9)*2.306*10 <sup>4</sup> *4.1855/1000;	
	-125.49712
	-873.50163

-212.61416

-53.32622

0.

13.44344

-596.18398

-652.68932

-37.29810